RAIRO-Oper. Res. 60 (2026) 1269–1286
https://doi.org/10.1051/ro/2026031

RAIRO Operations Research
www.rairo-ro.org

SOME THREE-TERM RMIL CONJUGATE GRADIENT METHODS WITH DESCENT PROPERTY FOR SOLVING OPTIMIZATION PROBLEMS WITH APPLICATION

Nasiru Salihu[1,2,*], Poom Kumam[1], Suraj Salihu[3] and Thidaporn Seangwattana[4]

Abstract. To propose a conjugate gradient (CG) scheme with a promising structure, it has been observed that Newton's direction is optimal when the current iteration is near the solution, and the objective function behaves like a quadratic. However, for large-scale problems, a method that does not require second-derivative information is often necessary. Therefore, to develop a more effective scheme for handling complex problems, we apply the standard secant equation to construct a combination of three-term CG search directions using the β_k^{RMIL} method. This combination approximates the quasi-Newton direction and ensures sufficient descent. Furthermore, we establish the global convergence of the scheme under mild assumptions, demonstrating that the algorithm is robust and reliable compared to earlier CG methods. Finally, we showcase the efficiency of the proposed scheme by applying it to solve a three degrees of freedom (3DOF) motion control model.

Mathematics Subject Classification. 65K05, 90C06, 90C30, 90C47, 90C90.

Received October 18, 2023. Accepted March 26, 2026.

1. Introduction

In this paper, we consider an optimization model of the form

$$\min f(x), \qquad x \in \mathbb{R}^n, \tag{1.1}$$

where the variable x is unconstrained, and the function $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be smooth, meaning its gradient, denoted as $g(x) = \nabla f(x)$, is computable. The optimization problem in (1.1) arises in various scientific and engineering applications [8,10,38,43]. These problems can be tackled using iterative computational

Keywords. Three term conjugate gradient method, conjugacy condition, Newton direction, global convergence, motion control.

[1] Center of Excellence in Theoretical and Computational Science (TaCS-CoE), Fixed Point Research Laboratory, Fixed Point Theory and Applications Research Group, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok 10140, Thailand.

[2] Department of Mathematics, Faculty of Sciences, Modibbo Adama University, Yola 652105, Nigeria.

[3] Department of Computer Science, Faculty of Science, Gombe State University, Gombe, Nigeria.

[4] Faculty of Science Energy and Environment, King Mongkut's University of Technology North Bangkok, Rayong Campus (KMUTNB), Rayong 21120, Thailand.

*Corresponding author: nsalihu@mau.edu.ng, nasirussalihu@gmail.com

techniques, including first-order descent methods and second-order methods [3]. Among first-order methods, conjugate gradient (CG) techniques are particularly attractive due to their strong theoretical properties, low memory requirements, and ease of implementation [28]. One notable variant is the three-term CG method, which generates a descent search direction by incorporating an additional term $\theta_k y_{k-1}$ into the classical CG search direction. This approach iteratively constructs a sequence $\{x_k\}$, starting from an initial estimate $x_0 \in \mathbb{R}^n$ and following the update rule:

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k = 0, 1, 2, \dots. \tag{1.2}$$

The scalar parameter $\alpha_k > 0$, which is referred to as the step size, is calculated to approximately satisfy specific predefined rules [19]. Furthermore, the step size is determined along the search direction d_k, which is defined as follows:

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} - \theta_k y_{k-1}, & \text{if } k \geq 1. \end{cases} \tag{1.3}$$

The scalar coefficients β_k and θ_k play a crucial role in determining the specific variant of the CG method. Several well-known choices for β_k have been introduced in the literature, including those by Fletcher and Reeves [16], Dai and Yuan [9], Fletcher [15], Hestenes and Stiefel [21], Polak *et al.* [31, 32], as well as Liu and Storey [23]. These methods define β_k using the following respective formulas:

$$\beta_k^{\mathrm{FR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \beta_k^{\mathrm{DY}} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{\mathrm{CD}} = -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}, \tag{1.4}$$

$$\beta_k^{\mathrm{HS}} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{\mathrm{PRP}} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \quad \beta_k^{\mathrm{LS}} = -\frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}, \tag{1.5}$$

where $y_{k-1} = g_k - g_{k-1}$, and $\|\cdot\|$ denotes the standard ℓ_2-norm.

Theoretically, it has been observed that methods incorporating $\|g_k\|^2$ in their design exhibit stable theoretical properties. However, their computational performance often yields modest results [3]. In contrast, methods that include $g_k^T y_{k-1}$ in their formulation automatically perform a restart when jamming a problem occurs, meaning the algorithm moves slowly without making significant progress. This characteristic enhances their numerical effectiveness [28]. Nevertheless, their global convergence is not always stable and requires certain variants of Wolfe-type line search rules to ensure convergence [37].

Despite the fact that the PRP method is one of the most reliable CG methods, Powell [33] argued that it exhibits an unstable global convergence property [37]. As a result, various researchers have proposed modifications to improve its stability, as seen in references [4, 17], among others. Similarly, Rivaie *et al.* [35] proposed a revised PRP variant given by

$$\beta_k^{\mathrm{RMIL}} = \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2}. \tag{1.6}$$

For this method, convergence requires the assumption $g_k^T g_{k-1} \geq 0$, which does not always hold. To address this issue, Dai [6] proposed a correction and reformulated (1.6) as

$$\beta_k^{\mathrm{RMIL+}} = \max\{\beta_k^{\mathrm{RMIL}}, 0\}. \tag{1.7}$$

The method has received considerable attention recently. However, it is also affected by convergence issues [36]. As a result, researchers have shown particular interest in modifying the schemes in (1.5) by introducing three-term CG methods to improve their convergence properties. For instance, a three-term PRP (TTPRP) CG scheme was proposed in [46]. This method is globally convergent using a modified version of the Armijo line search.

Similarly, using the Gram–Schmidt orthogonalization technique, Cheng [5] introduced a three-term PRP scheme that achieves global convergence by adopting the strong Wolfe line search technique [44]. In a similar manner, Zhang *et al.* [47] presented a three-term HS method, which also attains global convergence through the application of the standard Wolfe line search rule. Building on these advancements, Narushima *et al.* [27] extended the idea and proposed a range of three-term CG methods, encompassing the aforementioned methods as specific cases. These methods exhibit significant theoretical advantages [24].

The methods described above generate a search direction d_k, that satisfies the sufficient descent criterion even without considering a line search [25]. However, these methods do not satisfy the well-known conjugacy condition, which is equally important for the convergence of a CG method. Recently, similar three-term CG methods have been proposed using β_k^{LS} and β_k^{RMIL}, which are globally convergent and ensure sufficient descent under certain variants of the Wolfe line search criterion, as demonstrated by Liu *et al.* [25] and Sulaiman *et al.* [44], respectively.

Some standard CG schemes in [13,20,40,45] are primarily derived from the following pure conjugacy condition:

$$d_k^T y_{k-1} = 0. \tag{1.8}$$

This condition is crucial for both the convergence of a CG method and its numerical performance. However, the effectiveness of these algorithms is largely dependent on the precision of the line search [3]. In practice, the step size is typically approximated by using inexact line search rules. Consequently, more generalized forms of this condition were proposed by Perry [30] and Dai and Liao [7]:

$$d_k^T y_{k-1} = -g_k^T s_{k-1}, \tag{1.9}$$

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \qquad t \geq 0. \tag{1.10}$$

This criterion has been used in the development of CG schemes in [14, 22, 39, 41, 42].

Furthermore, to propose a CG scheme with a promising structure, it has been observed that Newton's direction is the optimal search direction when the current iteration is close to the solution and the objective function behaves like a quadratic function [1]. However, for large-scale problems, methods that do not require second-derivative information are often preferred [2]. Inspired by these discussions, we constructed a three-term CG search direction based on the β_k^{RMIL} method, ensuring sufficient descent while approximating the quasi-Newton direction. Here are some key contributions of this paper:

- Two three-term CG methods are introduced: one based on the RMIL CG method using the Gram–Schmidt orthogonalization procedure, and the other as a convex combination with the CG method suggested by Liu *et al.* [24].
- The search directions are consistently descending without requiring any line search in each iteration.
- The algorithm is proven to achieve global convergence under certain standard assumptions.
- The new convex combination of the three-term CG method is applied to a time-varying optimization problem involving a robotic model.

The remainder of the paper is structured as follows:

The next section presents the new TTCG method and its design. Section 3 analyzes the global convergence results of the new scheme. Section 4 discusses the experimentation and comparison. Finally, the conclusion is provided in the last section.

2. Descent property of RMIL methods and algorithm

We begin by describing the motivation behind the formulation of the two three-term CG methods and the algorithm. Recently, Liu *et al.* [24] presented a three-term RMIL CG search direction:

$$d_k^{\mathrm{RMIL1}} = -g_k + \beta_k^{\mathrm{RMIL}} d_{k-1} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} y_{k-1}, \tag{2.1}$$

where they demonstrated the method's convergence by applying the standard Wolfe line search criteria.

To suggest the second three-term direction, we employ the Gram–Schmidt orthogonalization procedure on d_{k-1} and g_k as follows:

$$d_k^{\mathrm{RMIL2}} = -g_k + \beta_k^{\mathrm{RMIL}} d_{k-1} - \beta_k^{\mathrm{RMIL}} \frac{g_k^T d_{k-1}}{\|g_k\|^2} g_k. \tag{2.2}$$

This, in conjunction with the three-term direction in (2.1), is used to propose the new three-term CG method.

Before presenting the new method, consider pre-multiplying (2.1) and (2.2) by g_k^T :

$$g_k^T d_k = -\|g_k\|^2 + \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} g_k^T d_{k-1} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} g_k^T y_{k-1} = -\|g_k\|^2,$$
$$g_k^T d_k = -\|g_k\|^2 + \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} g_k^T d_{k-1} - \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} \frac{g_k^T d_{k-1}}{\|g_k\|^2} \|g_k\|^2 = -\|g_k\|^2.$$

This implies that the two search directions generated are sufficiently descent without using a line search rule, *i.e.*,

$$g_k^T d_k = -\|g_k\|^2. \tag{2.3}$$

Applying Cauchy–Schwartz inequality, we obtain

$$\|d_k\| \ \geq \ \|g_k\| \ \Rightarrow \ \frac{1}{\|d_k\|} \ \leq \ \frac{1}{\|g_k\|}. \tag{2.4}$$

However, the formulations above do not fulfil the conjugacy condition. Thus, considering the nice convergence property of Newton method, it is crucial to formulate a CG method by combining d_k^{RMIL1} and d_k^{RMIL2} as:

$$d_k^{\mathrm{RMIL3}} = (1-\eta_k) d_k^{\mathrm{RMIL1}} + \eta_k d_k^{\mathrm{RMIL2}}, \tag{2.5}$$

where η_k is called hybridization that characterize the choice of the CG method. Now, using (2.1) and (2.2) in (2.5) and after some simple simplifications, we have

$$d_k^{\mathrm{RMIL3}} = d_k^{\mathrm{RMIL1}} + \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \left(y_{k-1} - \frac{g_k^T y_{k-1}}{\|g_k\|^2} g_k \right), \tag{2.6}$$

or more precisely,

$$d_k^{\mathrm{RMIL3}} = d_k^{\mathrm{RMIL1}} + \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \cdot P_k \cdot y_{k-1}, \tag{2.7}$$

where P_k is the orthogonal projection and symmeric non-negative definite matrix with $\|P_k\| = 1$ and is given by

$$P_k = I - \frac{g_k g_k^T}{\|g_k\|^2}. \tag{2.8}$$

Next, we explain the motivation behind the definition of η_k and how to get the required value of the parameter. In quasi-Newton (QN) method, the matrix B_{k-1} approximates $\nabla^2 f(x_{k-1})$, the Hessian matrix is updated in such a way that the new approximate matrix B_k satisfies suitable secant equation, for instance,

$$B_k s_{k-1} = y_{k-1}. \tag{2.9}$$

To obtain a CG method with less memory requirement, the quasi-Newton direction given by $d_k^{QN} = -B_k^{-1} g_k$ is needed. Therefore, using y_{k-1}^T, s_{k-1} and d_k^{QN} with (2.6) and (2.9), we can justify the choice of the parameter η_k, *i.e.*,

$$-B_k^{-1} g_k = d_k^{\mathrm{RMIL1}} - \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \left(\frac{g_k^T y_{k-1}}{\|g_k\|^2} g_k - y_{k-1} \right)$$

$$
\begin{aligned}
-s_{k-1}^T g_k &= s_{k-1}^T B_k d_k^{\mathrm{RMIL1}} - \eta_k s_{k-1}^T B_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\left(\frac{g_k^T y_{k-1}}{\|g_k\|^2} g_k - y_{k-1}\right) \\
&= -g_k^T y_{k-1} + \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} d_{k-1}^T y_{k-1} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\|y_{k-1}\|^2 \\
&\quad - \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\left(\frac{(g_k^T y_{k-1})^2}{\|g_k\|^2} - \|y_{k-1}\|^2\right).
\end{aligned} \tag{2.10}
$$

Since $y_{k-1} = g_k - g_{k-1}$, using (2.3) and squaring both sides of (2.4), we get

$$
\begin{aligned}
-s_{k-1}^T g_k &= -g_k^T y_{k-1} + \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} d_{k-1}^T g_k - \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} d_{k-1}^T g_{k-1} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\|y_{k-1}\|^2 \\
&\quad - \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\left(\frac{(g_k^T y_{k-1})^2}{\|g_k\|^2} - \|y_{k-1}\|^2\right) \\
&= -g_k^T y_{k-1} + \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} d_{k-1}^T g_k + \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2}\|g_{k-1}\|^2 - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\|y_{k-1}\|^2 \\
&\quad - \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\left(\frac{(g_k^T y_{k-1})^2}{\|g_k\|^2} - \|y_{k-1}\|^2\right) \\
&\le \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} d_{k-1}^T g_k - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\|y_{k-1}\|^2 - \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\left(\frac{(g_k^T y_{k-1})^2}{\|g_k\|^2} - \|y_{k-1}\|^2\right).
\end{aligned} \tag{2.11}
$$

Finding the η value at which the relation (2.11) holds with equality gives

$$
-s_{k-1}^T g_k = \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2} g_k^T d_{k-1} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\|y_{k-1}\|^2 + \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}\left(\|y_{k-1}\|^2 - \frac{(g_k^T y_{k-1})^2}{\|g_k\|^2}\right). \tag{2.12}
$$

Clearly, the pure conjugacy condition holds automatically from equation (2.12) when $g_k^T d_{k-1} = 0$. Now, let us examine the scenario where $g_k^T d_{k-1} \neq 0$. Obviously, since $s_{k-1} = \alpha_{k-1} d_{k-1}$, we obtain the hybridization parameter as

$$
\eta_k = \frac{\|y_{k-1}\|^2 - g_k^T y_{k-1} - s_{k-1}^T d_{k-1}}{\|y_{k-1}\|^2 - \left(\frac{g_k^T y_{k-1}}{\|g_k\|}\right)^2}. \tag{2.13}
$$

Next, the parameter η_k is well defined when the denominator is nonzero, which leads to the condition:

$$
\|y_{k-1}\|^2 \neq \left(\frac{g_k^T y_{k-1}}{\|g_k\|}\right)^2.
$$

This ensures that the numerator remains finite and avoids division by zero. Therefore, η_k is well-defined when y_{k-1} is not collinear with g_k. To take advantage of the rapid local convergence, Dai's modification of the RMIL method in (1.7) addresses the case where $g_k^T y_{k-1} > 0$. In particular, $\beta_k^{\mathrm{RMIL}} > 0$, and $\beta_k^{\mathrm{RMIL}} = 0$ can be interpreted as its truncated form, which helps mitigate potential issues. In order to counteract the potential unboundedness of η_k and ensure global convergence, we propose the following strategy to build the search direction:

$$
d_k = \begin{cases} -g_k, & \text{if } g_k^T y_{k-1} \le 0, \\ d_k^{\mathrm{RMIL3}} & \text{otherwise.} \end{cases} \tag{2.14}
$$

So that the search direction d_k^{RMIL3} is given by (2.6) and for convenience we refine η_k as

$$
\eta_k = \begin{cases} \frac{\|y_{k-1}\|^2 - g_k^T y_{k-1} - t s_{k-1}^T d_{k-1}}{\|g_k\|^2 \|y_{k-1}\|^2 - (g_k^T y_{k-1})^2} . \|g_k\|^2, & \text{if } k \in K, \\ 0, & \text{if } k \in \mathbb{N} \setminus K. \end{cases} \tag{2.15}
$$

Formula (2.15) introduces the index set K to refine the search direction d_k^{RMIL3} based on the relative alignment of the current gradient g_k and the previous gradient g_{k-1}. The parameter t, known as the Dai and Liao parameter, plays a crucial role in ensuring conjugacy, as defined by equation (1.10). The index set K contains the indices where the gradient g_k and the previous gradient g_{k-1} are sufficiently aligned, as determined by the condition:

$$K = \left\{ k \in \mathbb{N} \;\middle|\; 0 < \frac{g_k^T y_{k-1}}{\|g_k\| \|y_{k-1}\|} \leq (1-\varphi) \right\}, \tag{2.16}$$

where $\varphi \in (0,1)$ is a constant. To prevent division by zero in (2.15), we use the condition from (2.16):

$$\frac{g_k^T y_{k-1}}{\|g_k\| \|y_{k-1}\|} \;\leq\; (1-\varphi). \tag{2.17}$$

Since $\varphi \in (0,1)$, it follows that $\frac{1}{1-\varphi} > 1$, which implies

$$g_k^T y_{k-1} \;<\; \frac{g_k^T y_{k-1}}{1-\varphi}.$$

Using (2.17), we further obtain

$$\frac{g_k^T y_{k-1}}{1-\varphi} \;\leq\; \frac{g_k^T y_{k-1}}{\|g_k\| \|y_{k-1}\|}.$$

This leads to the inequality

$$\|g_k\|^2 \|y_{k-1}\|^2 > (g_k^T y_{k-1})^2.$$

Thus, the condition ensures that updates occur only when the gradients are not significantly misaligned. As expected, the number of elements in K is typically small, as it corresponds to cases where the gradients are closely aligned. This selection helps focus the search on regions where gradient information remains valuable, improving efficiency and convergence.

Remark 2.1. In the development of CG algorithms, the conjugacy condition plays a vital role in ensuring efficiency. Conjugate directions allow a convex quadratic function to be minimized over their spanned subspace by sequentially minimizing along each direction. This leads to faster convergence compared to standard gradient descent. In particular, the relation (2.12) satisfies Perry's condition (1.9), given by $d_k^T y_{k-1} = -g_k^T s_{k-1}$. As a result, the extended Dai–Liao condition (1.10) is preferable for improved performance. It should be noted that when $t = 0$, the scaled version of the conjugacy condition is recovered. However, previous three-term RMIL methods in [24,44] do not incorporate this approach. Moreover, the methods in [5,24,25,27,44] do not explore the combination of two search directions in the design of a CG method. This observation motivated the development of the RMIL method presented in (2.14)–(2.16).

Next, we the implementation procedure of the proposed method described below:

Algorithm 1: NTTRMIL method.

Step 1: Given an initial point $x_1 \in \mathbb{R}^n$. Select $t > 0$, $\varphi \in (0,1)$, $0 < \delta < \sigma < 1$.
Step 2: If $\|g_k\| \leq \epsilon$, then stop. Otherwise.
Step 3: Compute $\alpha_k > 0$ satisfying

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k, \tag{2.18}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{2.19}$$

where $0 < \delta < \sigma < 1$.
Step 5: Set $x_{k+1} = x_k + \alpha_k d_k$.
Step 6: Compute d_k by (2.6), (2.14) and (2.15).
Step 7: Update the next iterate from Step 2.

Remark 2.2. The analysis of the proposed scheme implies that the search direction d_k satisfies the sufficient descent condition for the objective function f at the point x_k without requiring a line search. Furthermore, the inequality $\|d_k\| \geq \|g_k\|$ ensures that β_k^{RMIL} is positive as long as $\|g_k\| \neq 0$. This implies that the proposed scheme is well-defined.

3. Convergence analysis

To establish the global convergence of Algorithm 1, we impose certain assumptions on the objective function f.

Assumption 3.1. 1. Denoting $\Gamma = \{x \in \mathbb{R}^n : f(x_0) \geq f(x)\}$ as the level set, where x_1 is the initial point and bounded below such that

$$\|x\| \leq B, \quad \forall x \in \Gamma_0,\ B > 0. \tag{3.1}$$

2. Denoting Γ_0 as some neighborhood of Γ, and function f in Algorithm 1 is smooth with its gradient is Lipschitz continuous satisfying

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \Gamma_0, \quad L > 0. \tag{3.2}$$

Under these assumptions, it easy to see that

$$\|g_k\| \leq \gamma, \tag{3.3}$$

for all $x \in \Gamma_0, \quad \gamma > 0$.

The following result from [49] is an important part of the convergence analysis of the CG method.

Theorem 3.2. *Suppose that Assumption 3.1 holds. Consider the CG method of the form (1.2), where the search direction d_k is descent and α_k satisfies the strong Wolfe conditions. Then,*

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{3.4}$$

Following the approach in [17], we begin by examining some characteristics of d_k and the step s_k. Additionally, we consider a general objective function to establish a weaker convergence result.

Suppose that the sequence of iterations $\{x_k\}$ is generated by Algorithm 1. Then, we have

$$\liminf_{k \to \infty} \|g_k\| = 0, \tag{3.5}$$

where we assume, for contradiction, that there exists a positive constant $\rho > 0$ such that

$$\|g_k\| \geq \rho, \quad \forall\ k \in \mathbb{N}; \tag{3.6}$$

otherwise a stationary point has achieved.

Theorem 3.3. *Suppose that (3.5) and Assumption 3.1 hold. Let the sequences $\{x_k\}$ be generated by Algorithm 1. Then, there exist $C > 0$ and $M > 0$ such that $|\beta_k^{\mathrm{RMIL}}| \leq C\|s_{k-1}\|$ and*

$$\|T_k\| \leq M, \quad \forall\ k \in \mathbb{N}, \tag{3.7}$$

where

$$T_k = \begin{cases} -g_k, & \text{if } \ k = 1 \ \text{ or } \ g_k^T y_{k-1} \leq 0, \\ -g_k - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} y_{k-1} + \eta_k \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \left(y_{k-1} - \frac{g_k^T y_{k-1}}{\|g_k\|^2} g_k \right), & \text{otherwise.} \end{cases} \tag{3.8}$$

Proof. From (2.4), (3.6) with Lipschitz criterion, we have

$$|\beta_k^{\text{RMIL}}| \leq \frac{\|g_k\|\|y_{k-1}\|}{\|d_{k-1}\|^2} \leq \frac{L\gamma\|s_{k-1}\|}{\|g_{k-1}\|^2} \leq \frac{L\gamma}{\rho^2}\|s_{k-1}\|. \tag{3.9}$$

Setting $C = \frac{L\gamma}{\rho^2}$, the result is established. Similarly, using $\alpha_k d_k = s_k \ \Rightarrow d_k = \frac{s_k}{\alpha_k}$, (2.15) and (2.16), we have

$$\begin{aligned}
|\eta_k| &= \frac{|\|y_{k-1}\|^2 - g_k^T y_{k-1} - t s_{k-1}^T d_{k-1}|}{\|y_{k-1}\|^2 - \left(\frac{g_k^T y_{k-1}}{\|g_k\|}\right)^2} \\
&\leq \frac{\|y_{k-1}\|^2 + |g_k^T y_{k-1}| + t|s_{k-1}^T d_{k-1}|}{\frac{\|g_k\|^2\|y_{k-1}\|^2 - (g_k^T y_{k-1})^2}{\|g_k\|^2\|y_{k-1}\|^2}} \\
&= \frac{\|y_{k-1}\|^2 + |g_k^T y_{k-1}| + t|s_{k-1}^T d_{k-1}|}{1 - \left(\frac{g_k^T y_{k-1}}{\|y_{k-1}\|\|g_k\|}\right)^2} \\
&\leq \frac{\|y_{k-1}\|^2 + \|g_k\|\|y_{k-1}\| + t\frac{\|s_{k-1}\|^2}{\alpha_{k-1}}}{1-(1-\varphi)^2} \\
&\leq \frac{L^2\|s_{k-1}\|^2 + L\gamma\|s_{k-1}\| + t\frac{\|s_{k-1}\|^2}{\alpha_{k-1}}}{1-(1-\varphi)^2} \\
&\leq \frac{4L^2B^2 + 2L\gamma B + t\frac{4B^2}{\gamma_{k-1}}}{1-(1-\varphi)^2} = M_\eta,
\end{aligned} \tag{3.10}$$

since $\alpha \in (1,0)$, the conclusion follows easily. Therefore, from (2.4), (2.8), (3.8), (3.10) and $\|P_k\| = 1$, we have

$$\begin{aligned}
\|T_k\| &\leq \|g_k\| + \frac{\|g_k\|\|d_{k-1}\|}{\|d_{k-1}\|^2}\|y_{k-1}\| + |\eta_k|\frac{\|g_k\|\|d_{k-1}\|}{\|d_{k-1}\|^2}\|y_{k-1} - \frac{g_k^T y_{k-1}}{\|g_k\|^2}g_k\| \\
&\leq \|g_k\| + \frac{\|g_k\|}{\|d_{k-1}\|}\|y_{k-1}\| + |\eta_k|\frac{\|g_k\|}{\|d_{k-1}\|}\|P_k\|\|y_{k-1}\| \\
&\leq \|g_k\| + \frac{\|g_k\|}{\|g_{k-1}\|}\|y_{k-1}\| + |\eta_k|\frac{\|g_k\|}{\|g_{k-1}\|}\|y_{k-1}\| \\
&\leq \gamma + (1+|\eta_k|)\frac{\|g_k\|}{\|g_{k-1}\|}\|y_{k-1}\| \\
&\leq \gamma + L(1+M_\eta)\frac{\gamma}{\rho}\|s_{k-1}\| \\
&\leq \gamma + 2L(1+M_\eta)\frac{\gamma}{\rho}B.
\end{aligned} \tag{3.11}$$

Setting $M = \gamma + 2L(1+M_\eta)\frac{\gamma}{\rho}B$, we also get $\|T_k\| \leq M$ as claimed. □

Theorem 3.4. *Suppose that Assumption 3.1 holds and the sequences* $\{x_k\}$ *and* $\{d_k\}$ *are generated by the NTTRMIL method. The step-size* γ_k *satisfies the strong Wolfe line search conditions. If* (3.6) *holds, then, we have* $d_k \neq 0$ *and*

$$\sum_{k=1}^{\infty}\|u_k - u_{k-1}\|^2 < \infty. \tag{3.12}$$

Proof. It is obvious that $\|d_k\| \geq \|g_k\| \neq 0$ and this shows that u_k is unitary and well-defined. Now, let define $r_k = \frac{T_k}{\|d_k\|}$ and

$$\delta_k = \beta_k^{\text{RMIL}} \frac{\|d_{k-1}\|}{\|d_k\|}. \tag{3.13}$$

Using relations (3.8) and (3.13), we obtain

$$u_k = r_k + \delta_k u_{k-1}. \tag{3.14}$$

Since $\|u_k\| = \|u_{k-1}\| = 1$, it is not difficult to obtain

$$\|r_k\| = \|u_k - \delta_k u_{k-1}\| = \|u_{k-1} - \delta_k u_k\|. \tag{3.15}$$

The last equality with condition $\delta_k \geq 0$, implies

$$\begin{aligned} \|u_k - u_{k-1}\| &\leq \|(1+\delta_k)u_k - (1+\delta_k)u_{k-1}\| \\ &\leq \|u_k - \delta_k u_{k-1}\| + \|u_{k-1} - \delta_k u_k\| \\ &= 2\|r_k\|. \end{aligned} \tag{3.16}$$

Considering (3.6), and the Zoutendijik criteria, we conclude that

$$\begin{aligned} \sum_{k=1}^{\infty} \|r_k\|^2 = \sum_{k=1}^{\infty} \frac{\|T_k\|^2}{\|d_k\|^2} &\leq \sum_{k=1}^{\infty} \frac{M^2}{\|d_k\|^2} \\ &= \sum_{k=1}^{\infty} \frac{M^2}{\|g_k\|^4} \frac{\|g_k\|^4}{\|d_k\|^2} \\ &\leq \frac{M^2}{\rho^4} \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \end{aligned} \tag{3.17}$$

Clearly, this contradicts (3.16). Thus, the result must hold. Similarly, the implications of Theorems 3.3 and 3.4, implies we have established the convergence of NTTRMIL method. □

Theorem 3.5. *Suppose Assumption 3.1 holds and the sequence $\{x_k\}$ be generated by Algorithm 1. Then NTTR-MIL converges such that (3.5) holds.*

Proof. The proof follows from [29], which is divided into two steps.

I. A bound on the steps s_k. Let $\triangle \in \mathbb{N}$ is chosen large enough such that $\triangle \geq 4BC$, where B and C are define using (3.1). For any $l > k > k_0$ with $l - k \leq \triangle$, following the same method of proof as the case two of Theorem 3.2 in [29], we obtain

$$\sum_{j=k}^{l-1} \|s_k\| \leq 2B. \tag{3.18}$$

II. A bound on d_k, and therefore utilizing (3.9) and (3.11), we get

$$\begin{aligned} \|d_k^{\text{RMIL3}}\|^2 &\leq (\|T_k\| + |\beta_k^{\text{RMIL}}| + \|d_{k-1}\|)^2 \\ &\leq (M + C\|s_{k-1}\| + \|d_{k-1}\|)^2 \\ &\leq 2M^2 + 2(C\|s_{k-1}\| + \|d_{k-1}\|)^2. \end{aligned} \tag{3.19}$$

The conclusion of the proof is similar to Theorem 5 in [12] which is originally case (iii) of Theorem 3.2 in [18]. Thus, omit here and the proof is complete. □

4. Numerical results

In this section, we evaluate the performance of the NTTRMIL method by comparing it with the TTRMIL method [24], the TTRMIL+ method [44], and the TTPRP method [46] using a set of 132 test problems collected from [3] and [26]. These problems are detailed in Table 1. The MATLAB version 9.12 code was executed on a Dell laptop with a Core i7-1195G7 processor, 16 GB of RAM, and a 2.90 GHz CPU. During execution, we configure specific parameters, $\delta = 10^{-4}$ and $\sigma = 10^{-3}$, in accordance to the strong Wolfe line search conditions for all the algorithms, with a stopping criterion of $\|g_k\| \leq 10^{-6}$.

To determine the optimal value for the parameter t in the NTTRMIL algorithm, we carried out tests with various selected values, including $\{0.0001, 0.01, 0.5, 1, 10, 20, 50, 100\}$ The best results were achieved with $t = 0.01$. Furthermore, we set $\varphi = 0.25$, so that the condition in (2.16) becomes:

$$0 < \frac{g_k^T y_{k-1}}{\|g_k\|\|y_{k-1}\|} \leq 0.75.$$

This means that only in cases where the gradients are sufficiently aligned, the search direction will be refined. We also compared the numerical results using the performance profiling method introduced by Dolan and Moré [11]. The experimental results are presented in Table 1 and are further analyzed and visually interpreted in Figures 1–3.

The interpretations show that the NTTRMIL method is efficient and preferable compared to other methods based on three metrics: the number of iterations (NI), the central processing unit time (CPU), and the number of function evaluations (FE). Therefore, the NTTRMIL method demonstrates the best performance by solving an average of 85% of the problems, with a 70% win rate, *i.e.*, it is the best solver for the given test functions. It is followed by the TTRMIL+, TTRMIL, and TTPRP methods, which have win rates of 75% to 21.7%, 73% to 20%, and 61% to 16%, respectively.

5. Application of NTTRMIL on real-time 3DOF robotic model

In this section, we demonstrate the use of Algorithm 1 in solving a robotic control problem, as discussed in [34]. The MATLAB code was executed with the same specifications as mentioned above. Now, suppose that the kinematics model at the position level of a planar manipulator is formulated as

$$g(\gamma) = \begin{bmatrix} a_1\cos(\gamma_1) + a_2\cos(\gamma_1+\gamma_2) + a_3\cos(\gamma_1+\gamma_2+\gamma_3) \\ a_1\sin(\gamma_1) + a_2\sin(\gamma_1+\gamma_2) + a_3\sin(\gamma_1+\gamma_2+\gamma_3) \end{bmatrix}. \tag{5.1}$$

Here, $g(\cdot)$ represents the kinematic mapping that connects the orientation and position of specific robot components. In this context, we denote the length of the ith segments as a_i (for $i = 1, 2, 3$), and the vector $\gamma \in \mathbb{R}^3$ within $g(\gamma)$ indicates the joint angles that describe the position of the end effector. Assuming that the vector $\delta_k \in \mathbb{R}^2$ represents the desired path at time $t_k \in [0, t_f]$, we can minimize the following optimization problem:

$$\min_{\gamma\in\mathbb{R}^3} \frac{1}{2}\|g(\gamma) - \delta_k\|^2, \tag{5.2}$$

where δ_k denotes the end effector controlled path at t_k of a Lissajous curve given by [48] as:

$$\delta_k = \begin{bmatrix} 1.5 + 0.4\sin\left(\frac{\pi t_k}{5}\right) \\ \frac{\sqrt{3}}{2} + 0.4\sin\left(\frac{\pi t_k}{5} + \frac{\pi}{3}\right) \end{bmatrix}. \tag{5.3}$$

The segments were initialized at the time instant $t = 0$, with $\gamma_0 = [\gamma_1, \gamma_2, \gamma_3] = [0, \frac{\pi}{3}, \frac{\pi}{2}]$, which are associated with the end effector vectors. The task period $[0, t_{\text{final}}]$ was partitioned into 200 parts, with the segments given by

TABLE 1. Numerical result of the methods.

	Test functions	TTRMIL			TTRMIL+			NTTRMIL			TTPRP		
S/N	Problems	NI	FE	CPU	NI	FE	CPU	NI	FE	CPU	NI	FE	CPU
1	EXTENDED PENALTY	9	95	0.0013	8	42	0.0069	9	93	0.0016	9	96	0.0010
2	EXTENDED PENALTY	6	82	0.0016	5	29	0.0044	7	86	0.0018	***	***	***
3	EXTENDED PENALTY	6	82	0.0020	5	29	0.0023	***	***	***	6	35	0.0021
4	EXTENDED MARATOS	***	***	***	***	***	***	3	121	0.0009	1215	24 101	0.0008
5	EXTENDED MARATOS	***	***	***	***	***	***	3	121	0.0018	521	9411	0.0020
6	DIAGONAL 5	2	11	0.0140	2	12	0.0210	2	13	0.0033	2	11	0.0018
7	DIAGONAL 5	2	11	0.0778	2	12	0.1088	2	13	0.1138	2	11	0.1093
8	DIAGONAL 5	2	11	0.1489	2	12	0.1943	2	13	0.1984	2	11	0.2173
9	TRECANNI	17	315	0.0003	26	385	0.0003	9	38	0.0011	37	773	0.0006
10	TRECANNI	38	883	0.0003	22	434	0.0004	7	32	0.0004	***	***	***
11	QUADRATIC PENALTY 1	12	197	0.0286	34	677	0.0320	8	34	0.0053	39	746	0.0571
12	QUADRATIC PENALTY 1	26	155	0.0114	19	176	0.0089	8	35	0.0032	22	409	0.0281
13	QUADRATIC PENALTY 1	***	***	***	***	***	***	***	***	***	***	***	***
14	QUADRATIC PENALTY 2	***	***	***	4	30	0.0197	***	***	***	***	***	***
15	QUADRATIC PENALTY 2	***	***	***	4	30	0.0037	***	***	***	***	***	***
16	QUADRATIC PENALTY 2	***	***	***	4	30	0.0027	***	***	***	***	***	***
17	QUADRATIC FUNCTION 1	57	1530	0.0839	40	829	0.0315	32	503	0.0203	73	2250	0.1003
18	QUADRATIC FUNCTION 1	57	1530	0.0689	40	829	0.0220	32	503	0.0208	73	2250	0.1027
19	QUADRATIC FUNCTION 1	65	1893	0.0870	38	769	0.0392	27	398	0.0133	59	1797	0.0782
20	QUADRATIC FUNCTION 2	***	***	***	4	17	0.0108	3	13	0.0034	55	1958	0.1380
21	QUADRATIC FUNCTION 2	***	***	***	4	17	0.0038	3	13	0.0034	***	***	***
22	QUADRATIC FUNCTION 2	***	***	***	4	17	0.0058	3	13	0.0096	***	***	***
23	POWER	15	61	0.0003	15	61	0.0002	13	53	0.0005	20	277	0.0004
24	POWER	17	69	0.0004	17	69	0.0002	15	61	0.0004	23	240	0.0005
25	ZETTL	158	3916	0.0003	368	4129	0.0003	10	154	0.0004	***	***	***
26	DIAGONAL 2	***	***	***	***	***	***	673	4347	0.7639	***	***	***
27	DIAGONAL 2	***	***	***	***	***	***	***	***	***	***	***	***
28	DIAGONAL 2	***	***	***	***	***	***	***	***	***	***	***	***
29	DIAGONAL 2	***	***	***	***	***	***	***	***	***	***	***	***
30	TEST	***	***	***	1015	7254	0.9716	15	272	0.0072	626	9378	0.3350
31	TEST	311	2262	0.1018	***	***	***	29	179	0.0048	642	9472	0.2942
32	SUM OF SQUARES	11	244	0.0005	7	119	0.0006	17	293	0.0002	16	428	0.0004
33	SUM OF SQUARES	17	506	0.0007	18	460	0.0011	12	301	0.0003	19	638	0.0008
34	SUM OF SQUARES	23	724	0.0013	24	719	0.0027	34	953	0.0006	22	754	0.0007
35	SHALLOW	280	7377	0.0005	841	11 713	0.0005	442	3858	0.0003	315	8461	0.0004
36	SHALLOW	283	7771	0.0013	180	2848	0.0014	7	236	0.0103	344	9391	0.0025
37	QUARTIC	3	24	0.0170	3	28	0.0163	3	29	0.0058	3	23	0.0060
38	QUARTIC	3	24	0.0115	3	28	0.0219	3	29	0.0086	3	23	0.0088
39	QUARTIC	5	93	0.1587	3	30	0.0675	3	35	0.0418	3	20	0.0312
40	QUARTIC	5	93	0.3463	3	30	0.1369	3	35	0.0830	5	94	0.3246
41	MATYAS	1	10	0.0018	1	10	0.0023	1	10	0.0005	1	10	0.0008
42	MATYAS	1	10	0.0006	1	10	0.0009	1	10	0.0008	1	10	0.0004
43	DIAGONAL 1	188	835	0.0928	205	1743	0.1620	123	1030	0.0529	207	2863	0.1174
44	DIAGONAL 1	188	835	0.0642	205	1743	0.1975	123	1030	0.0393	207	2863	0.1278
45	DIAGONAL 1	133	573	0.0468	169	1627	0.1804	116	886	0.0550	206	2963	0.1156
46	DIAGONAL 1	133	573	0.0457	169	1627	0.2147	116	886	0.0493	206	2963	0.1667
47	HAGER	***	***	***	***	***	***	***	***	***	***	***	***
48	HAGER	***	***	***	***	***	***	***	***	***	***	***	***
49	HAGER	***	***	***	***	***	***	3	113	0.0071	***	***	***
50	HAGER	***	***	***	***	***	***	4	165	0.0038	***	***	***
51	ZIRILLI	17	413	0.0017	10	139	0.0004	7	30	0.0003	***	***	***
52	RAYDAN 1	817	22 273	1.8696	842	18 223	0.9740	105	457	0.0270	***	***	***
53	RAYDAN 1	114	3214	0.1891	100	2482	0.2001	21	118	0.0073	122	3290	0.1769
54	RAYDAN 1	63	367	0.0268	71	541	0.0367	19	111	0.0107	123	2209	0.1126
55	RAYDAN 1	427	2393	0.2396	399	3950	0.2115	60	250	0.0153	837	13 446	0.7975
56	RAYDAN 2	2	9	0.0242	2	9	0.0231	2	9	0.0114	2	9	0.0198
57	RAYDAN 2	2	9	0.0605	2	9	0.0538	2	9	0.0319	2	9	0.0386
58	RAYDAN 2	2	9	0.1031	2	9	0.1265	2	9	0.0581	2	9	0.0674
59	FLETCHCER	2	11	0.0005	2	11	0.0004	2	11	0.0003	2	11	0.0003
60	FLETCHCER	3	15	0.0008	3	15	0.0008	2	11	0.0004	3	15	0.0004
61	FLETCHCER	3	15	0.0061	3	15	0.0045	3	15	0.0035	3	15	0.0045
62	DIAGONAL 3	92	2236	0.2808	122	2476	0.2206	32	151	0.0108	***	***	***
63	DIAGONAL 3	92	2236	0.1563	122	2476	0.1753	32	151	0.0085	***	***	***
64	DIAGONAL 3	12	150	0.0187	17	268	0.0299	8	96	0.0049	***	***	***
65	DIAGONAL 3	106	2486	0.1869	140	2964	0.1529	3	111	0.0096	***	***	***

TABLE 1. continued.

Test functions		TTRMIL			TTRMIL+			NTTRMIL			TTPRP		
S/N	Problems	NI	FE	CPU	NI	FE	CPU	NI	FE	CPU	NI	FE	CPU
66	EXTENDED DENSCHN B	15	159	0.0005	5	21	0.0007	6	34	0.0006	19	371	0.0007
67	EXTENDED DENSCHN B	29	656	0.0014	22	334	0.0019	5	21	0.0012	***	***	***
68	EXTENDED DENSCHN B	31	713	0.0021	22	334	0.0017	5	21	0.0012	***	***	***
69	DIAGONAL 6	2	12	0.0297	2	10	0.0255	2	11	0.0128	2	12	0.0170
70	DIAGONAL 6	2	12	0.0132	2	10	0.0089	2	11	0.0083	2	12	0.0084
71	DIAGONAL 6	2	12	0.0200	2	10	0.0149	2	11	0.0110	2	12	0.0167
72	DIAGONAL 6	2	12	0.0723	2	10	0.0598	2	11	0.0366	2	12	0.0595
73	DIAGONAL 4	19	469	0.0932	16	261	0.0565	21	85	0.0079	19	469	0.0618
74	DIAGONAL 4	23	583	0.5463	13	200	0.1605	21	85	0.0519	***	***	***
75	DIAGONAL 4	23	583	3.8566	19	322	3.1578	6	25	0.1113	***	***	***
76	DIAGONAL 7	4	66	0.0162	3	13	0.0099	3	13	0.0018	4	62	0.0049
77	DIAGONAL 7	4	66	0.0101	3	13	0.0028	3	13	0.0023	4	66	0.0192
78	DIAGONAL 7	4	66	0.0073	3	13	0.0041	3	13	0.0037	4	66	0.0054
79	DIAGONAL 8	2	10	0.0167	3	14	0.0177	3	14	0.0039	2	10	0.0017
80	DIAGONAL 8	2	10	0.0052	3	14	0.0071	3	14	0.0056	2	10	0.0030
81	DIAGONAL 9	***	***	***	***	***	***	1515	15 761	0.8092	***	***	***
82	DIAGONAL 9	***	***	***	***	***	***	3	112	0.0086	***	***	***
83	DENSCHN A	29	528	0.0011	55	1020	0.0016	3	115	0.0078	49	772	0.0009
84	DENSCHN A	31	585	0.0026	58	983	0.0028	3	115	0.0127	47	797	0.0029
85	DENSCHN C	3	159	0.0023	3	159	0.0015	2	106	0.0033	3	159	0.0016
86	DENSCHN C	3	159	0.0089	3	159	0.0075	2	106	0.0131	3	159	0.0125
87	EXTENDED BLOCK DIAGONAL 1	***	***	***	***	***	***	***	***	***	***	***	***
88	EXTENDED BLOCK DIAGONAL 1	***	***	***	***	***	***	***	***	***	***	***	***
89	EXTENDED BLOCK DIAGONAL 1	***	***	***	***	***	***	***	***	***	***	***	***
90	HIMMELBLAU	***	***	***	***	***	***	7	29	0.0003	36	546	0.0006
91	HIMMELBLAU	***	***	***	***	***	***	***	***	***	***	***	***
92	HIMMELBLAU	***	***	***	***	***	***	***	***	***	***	***	***
93	HIMMELBLAU	***	***	***	22	432	0.0024	***	***	***	***	***	***
94	DQDRTIC	407	11 904	0.0003	794	21 337	0.0008	335	8696	0.0003	898	26 746	0.0013
95	DQDRTIC	507	15 936	0.0012	1512	41 662	0.0024	222	6246	0.0013	***	***	***
96	DQDRTIC	1282	34 882	0.0003	1173	30 486	0.0009	226	5233	0.0004	1096	30 139	0.0007
97	DQDRTIC	935	23 069	0.0004	354	8936	0.0004	112	2546	0.0004	***	***	***
98	QUARTICM	2	10	0.0005	3	19	0.0007	4	39	0.0006	2	10	0.0025
99	QUARTICM	2	10	0.0018	3	19	0.0029	4	41	0.0017	2	10	0.0037
100	LINEAR PERTURBED	2	9	0.0005	2	9	0.0011	2	9	0.0005	2	9	0.0120
101	LINEAR PERTURBED	3	13	0.0008	3	13	0.0018	2	9	0.0010	3	13	0.0020
102	LINEAR PERTURBED	3	13	0.0016	3	13	0.0020	4	17	0.0012	3	13	0.0029
103	TWH	***	***	***	***	***	***	***	***	***	1397	27 492	0.0008
104	TWH	1238	5200	0.0003	***	***	***	***	***	***	1961	39 664	0.0004
105	ENGVAL1	16	167	0.0005	17	170	0.0004	12	133	0.0003	29	366	0.0004
106	ENGVAL1	36	785	0.0004	21	382	0.0005	11	148	0.0003	***	***	***
107	ENGVAL8	69	1671	0.0005	49	814	0.0006	33	139	0.0003	102	2471	0.0005
108	ENGVAL8	63	1658	0.0004	58	1514	0.0007	24	272	0.0004	75	2050	0.0004
109	DENSCHN F	31	594	0.0007	25	229	0.0006	17	327	0.0004	***	***	***
110	DENSCHN F	35	708	0.0018	26	233	0.0022	23	392	0.0015	***	***	***
111	DENSCHN F	35	708	0.0057	27	237	0.0126	25	433	0.0053	***	***	***
112	ARWHEAD	***	***	***	***	***	***	***	***	***	***	***	***
113	ARWHEAD	***	***	***	***	***	***	***	***	***	***	***	***
114	ARWHEAD	***	***	***	***	***	***	***	***	***	***	***	***
115	SIX HUMP	28	479	0.0006	26	429	0.0005	4	138	0.0003	***	***	***
116	SIX HUMP	26	504	0.0004	25	439	0.0005	10	55	0.0009	***	***	***
117	PRICE4	2	10	0.0003	2	10	0.0004	2	10	0.0004	2	10	0.0006
118	PRICE4	2	9	0.0003	2	9	0.0004	2	9	0.0002	2	9	0.0009
119	ZIRILLI	17	412	0.0005	10	140	0.0005	7	30	0.0003	***	***	***
120	ZIRILLI	17	410	0.0003	12	188	0.0007	7	30	0.0004	***	***	***
121	EXTENDED HIMMELBLAU	63	958	0.1734	46	1044	0.2935	49	1115	0.1753	86	3023	0.8819
122	EXTENDED HIMMELBLAU	63	955	0.1531	43	931	0.2158	50	1127	0.1794	86	3003	1.1930
123	EXTENDED HIMMELBLAU	69	1066	0.2752	48	1100	0.5174	50	1140	0.2582	92	3259	1.6155
124	ROTATED ELLIPSE	26	598	0.0004	40	910	0.0012	45	717	0.0004	36	1004	0.0004
125	ROTATED ELLIPSE	31	654	0.0004	53	1227	0.0004	51	748	0.0003	38	1058	0.0004
126	EL-ATTAR-VIDYASAGAR-DUTTA	31	495	0.0003	30	256	0.0003	3	125	0.0004	72	1879	0.0017
127	EL-ATTAR-VIDYASAGAR-DUTTA	47	257	0.0005	***	***	***	3	126	0.0002	56	1090	0.0006
128	EXTENDED HIEBERT	***	***	***	***	***	***	127	1658	0.0003	***	***	***
129	EXTENDED TRIDIAGONAL 1	9	49	0.0008	14	175	0.0004	56	326	0.0003	40	628	0.0010
130	EXTENDED TRIDIAGONAL 1	9	49	0.0006	15	179	0.0007	55	352	0.0004	122	1864	0.0010
131	THUMP	***	***	***	27	264	0.0004	3	135	0.0002	***	***	***
132	THUMP	17	337	0.0007	18	231	0.0010	8	62	0.0008	***	***	***

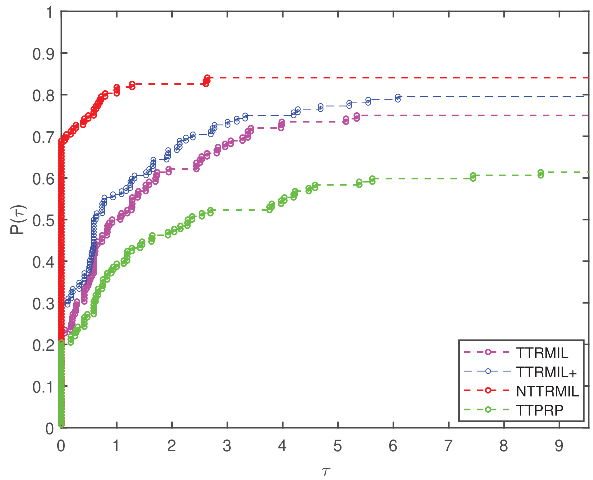

FIGURE 1. Number of iterations performance profiles of the methods.

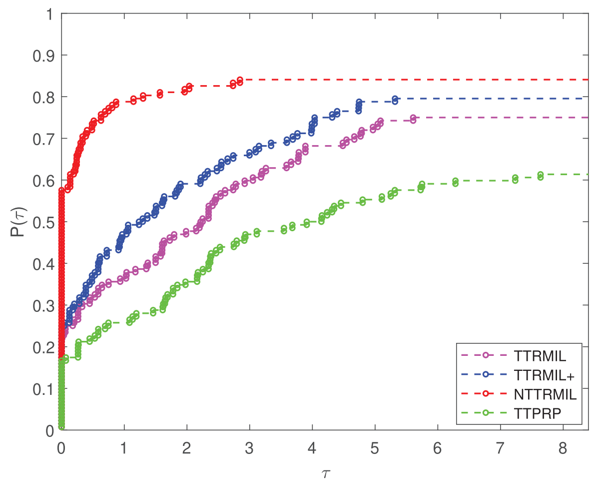

FIGURE 2. Number of function evaluation performance profiles of the methods.

a_i (for $i = 1, 2, 3$), and $t_{\text{final}} = 10$ s. The results of the motion control experiments are described in Figures 4–7. The interpretation of the results shows that the NTTRMIL method fulfills the desired task of synthesizing and passing through the required route, as seen in the robot trajectories plotted in Figures 6 and 7, with a minimum error of 10^{-8} as shown in Figures 4 and 5.

6. CONCLUSION

In this work, we proposed a novel three-term RMIL-based conjugate gradient method as a convex combination by considering the Gram–Schmidt orthogonalization procedure applied to d_{k-1} and g_k, in conjunction with the method by Liu *et al.* [24]. The method ensures sufficient descent without requiring a line search. We introduced a robust search direction definition that ensures the method is well-posed, even for the first iteration, when the previous gradient is undefined. Additionally, we utilized an indicator set K governed by φ, which adapts the search process based on gradient alignment. We clarified the impact of φ, noting that while a larger value

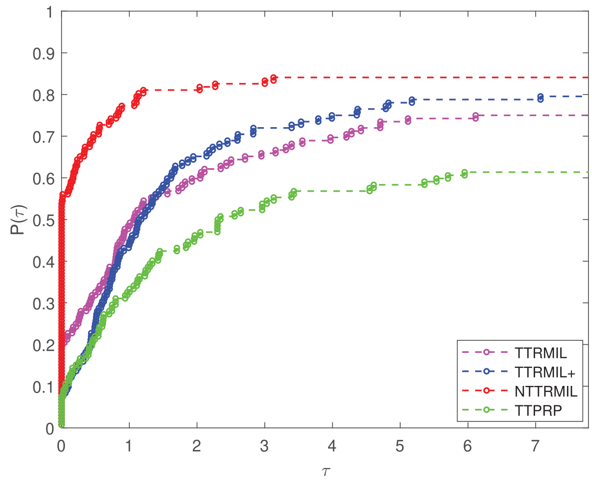

FIGURE 3. Time performance profiles of the methods.

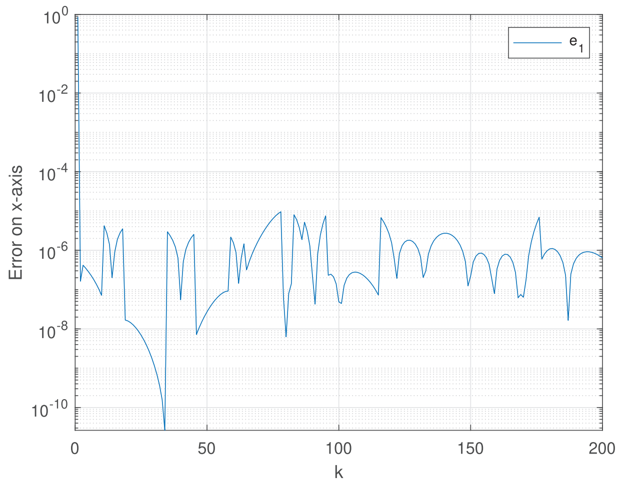

FIGURE 4. Error tracking by NTTRMIL on x-axis.

(*e.g.*, $\varphi = 0.25$) increases the likelihood of a non-empty K, extreme cases where K could still be empty remain possible. We also provided global convergence analysis under the strong Wolfe line search, demonstrating the method's theoretical soundness. Numerical experiments confirmed the effectiveness of the proposed approach, with $\varphi = 0.25$ chosen to balance stability and flexibility. Furthermore, promising numerical results for the NTTRMIL method were shown in comparison with earlier CG methods, especially when $t = 0.001$. Finally, the efficiency of the proposed method was demonstrated in solving a robotic motion control model.

This study contributes to the advancement of CG methods by improving their applicability to large-scale optimization problems without requiring second-order derivative information. The proposed method is particularly relevant for applications in engineering. By clearly identifying the practical and theoretical advantages, this work enhances the impact of the method within the optimization research community.

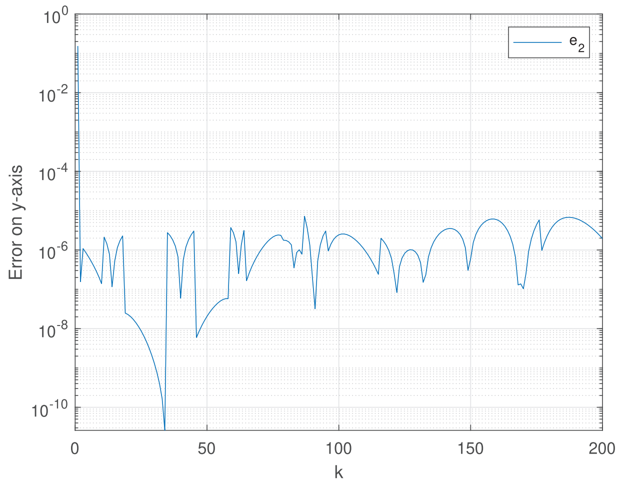

FIGURE 5. Error tracking by NTTRMIL on y-axis.

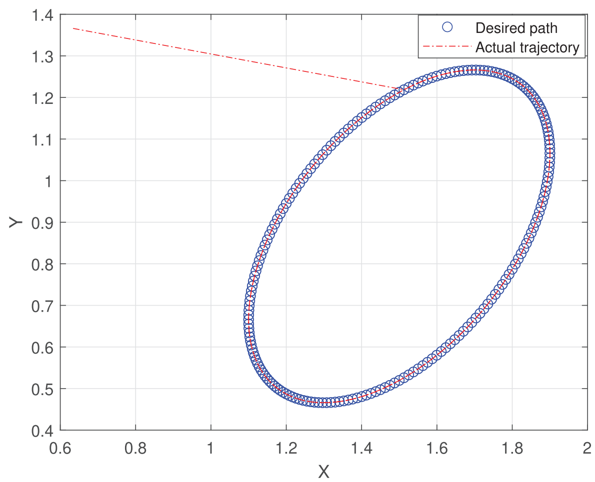

FIGURE 6. End effector of RMIL+ trajectory and desired path.

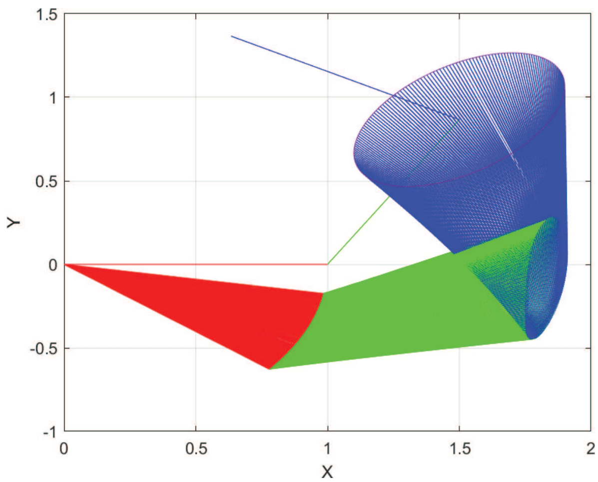

FIGURE 7. Robot trajectories synthesized by NTTRMIL.

ACKNOWLEDGMENTS

The authors acknowledge the support provided by the Petchra Pra Jom Klao PhD Scholarship of King Mongkut's University of Technology Thonburi (KMUTT) under Contract No. 52/2564, and the Center of Excellence in Theoretical and Computational Science (TaCS-CoE), KMUTT. Moreover, this research was funded by National Science, Research and Innovation Fund (NSRF), King Mongkut's University of Technology North Bangkok with Contract No. KMUTNB-FF-69-A-06.

FUNDING

The work of these authors is supported by NSRF with Contract No. KMUTNB- FF-69-A-06.

DATA AVAILABILITY STATEMENT

The codes of the current study are available upon request from the the corresponding author.

AUTHOR CONTRIBUTION STATEMENT

Nasiru Salihu and Suraj Salihu: Conceptualization, Methodology, Coding. Nasiru Salihu: Writing-Original draft. Nasiru Salihu, Poom Kumam and Thidaporn Seangwattana: Visualization, Investigation, Validation. Poom Kumam: Supervision. Poom Kumam and Thidaporn Seangwattana. Validating the Experiment. Nasiru Salihu, Poom Kumam, Suraj Salihu and Thidaporn Seangwattana: Writing- Reviewing and Editing the manuscript.

REFERENCES

[1] N. Andrei, A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes–Stiefel and Dai–Yuan. *Stud. Inform. Control.* **17** (2008) 57.

[2] N. Andrei, A double parameter scaled BFGS method for unconstrained optimization. *J. Comput. Appl. Math.* **332** (2018) 26–44.

[3] N. Andrei, Nonlinear Conjugate Gradient Methods for Unconstrained Optimization. Springer, Cham (2020).

[4] S. Babaie-Kafaki and R. Ghanbari, A descent family of Dai–Liao conjugate gradient methods. *Optim. Method. Softw.* **29** (2014) 583–591.

[5] W. Cheng, A two-term PRP-based descent method. *Numer. Funct. Anal. Optim.* **28** (2007) 1217–1230.

[6] Z. Dai, Comments on a new class of nonlinear conjugate gradient coefficients with global convergence properties. *Appl. Math. Comput.* **276** (2016) 297–300.

[7] Y.-H. Dai and L.-Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Optim.* **43** (2001) 87–101.

[8] Z. Dai and T. Wu, The impact of oil shocks on systemic risk of the commodity markets. *J. Syst. Sci. Complex.* **37** (2024) 1–24.
[9] Y.-H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **10** (1999) 177–182.
[10] Z. Dai and X. Zhang, Climate policy uncertainty and risks taken by the bank: evidence from China. *Int. Rev. Financ. Anal.* **87** (2023) 102579.
[11] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Math. Program.* **91** (2002) 201–213.
[12] X. Dong, H. Liu, Y. He, S. Babaie-Kafaki and R. Ghanbari, A new three–term conjugate gradient method with descent direction for unconstrained optimization. *Math. Model. Anal.* **21** (2016) 399–411.
[13] S.S. Đorđević, New hybrid conjugate gradient method as a convex combination of LS and CD methods. *Filomat* **31** (2017) 1813–1825.
[14] S.S. Đorđević, New hybrid conjugate gradient method as a convex combination of LS and FR methods. *Acta Math. Sci.* **39** (2019) 214–228.
[15] R. Fletcher, Practical Methods of Optimization. A Wiley Interscience Publication, New Jersey (1987).
[16] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients. *Comput. J.* **7** (1964) 149–154.
[17] J.C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.* **2** (1992) 21–42.
[18] W.W. Hager and H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **16** (2005) 170–192.
[19] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2** (2006) 35–58.
[20] N. Hamel, N. Benrabia, M. Ghiat and H. Guebbai, A new hybrid conjugate gradient algorithm based on the Newton direction to solve unconstrained optimization problems. *J. Appl. Math. Comput.* **69** (2023) 2531–2548.
[21] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving. *J. Res. Natl. Bur. Stand.* **49** (1952) 409.
[22] J. Liu and S. Li, New hybrid conjugate gradient method for unconstrained optimization. *Appl. Math. Comput.* **245** (2014) 36–43.
[23] Y. Liu and C. Storey, Efficient generalized conjugate gradient algorithms, part 1: theory. *J. Optim. Theory Appl.* **69** (1991) 129–137.
[24] J. Liu, Y. Feng and L. Zou, Some three-term conjugate gradient methods with the inexact line search condition. *Calcolo* **55** (2018) 1–16.
[25] J. Liu, Y. Zhao and X. Wu, Some three-term conjugate gradient methods with the new direction structure. *Appl. Numer. Math.* **150** (2020) 433–443.
[26] J. Momin and Y. Xin-She, A literature survey of benchmark functions for global optimization problems. *Int. J. Math. Model. Numer. Optim.* **4** (2013) 150–194.
[27] Y. Narushima, H. Yabe and J.A. Ford, A three-term conjugate gradient method with sufficient descent property for unconstrained optimization. *SIAM J. Optim.* **21** (2011) 212–230.
[28] S. Nasiru, R.O. Mathew, Y.W. Mohammed, S.H. Abubakar and S. Suraj, A Dai–Liao hybrid conjugate gradient method for unconstrained optimization. *IJIO* **2** (2021) 69–84.
[29] J. Nocedal, Theory of algorithms for unconstrained optimization. *Acta Numer.* **1** (1992) 199–242.
[30] A. Perry, A modified conjugate gradient algorithm. *Oper. Res.* **26** (1978) 1073–1078.
[31] E. Polak and G. Ribiere, Note sur la convergence de methodes de directions conjuguees. *USSR Comput. Math. Math. Phys.* **9** (1969) 94–112.
[32] B.T. Polyak, A general method for solving extremal problems. *Dokl. Akad. Nauk SSSR* **174** (1967) 33–36.
[33] M.J. Powell, Nonconvex minimization calculations and the conjugate gradient method, in Numerical Analysis. Springer, Berlin (1984) 122–141.
[34] A. Renfrew, Introduction to robotics: mechanics and control. *Int. J. Electr. Eng. Educ.* **41** (2004) 388.
[35] M. Rivaie, M. Mamat, L.W. June and I. Mohd, A new class of nonlinear conjugate gradient coefficients with global convergence properties. *Appl. Math. Comput.* **218** (2012) 11323–11332.
[36] N. Salihu, M. Odekunle, M. Waziri and A. Halilu, A new hybrid conjugate gradient method based on secant equation for solving large scale unconstrained optimization problems. *Iran. J. Optim.* **12** (2020) 33–44.
[37] N. Salihu, M.R. Odekunle, A.M. Saleh and S. Salihu, A Dai–Liao hybrid hestenes-stiefel and fletcher-revees methods for unconstrained optimization. *IJIO* **2** (2021) 33–50.
[38] N. Salihu, H.A. Babando, I. Arzuka and S. Salihu, A hybrid conjugate gradient method for unconstrained optimization with application. *Bangmod J-MCS* **9** (2023) 24–44.

[39] N. Salihu, P. Kumam, I.M. Sulaiman and T. Seangwattana, An efficient spectral minimization of the Dai–Yuan method with application to image reconstruction. *AIMS Math.* **8** (2023) 30940–30962.

[40] N. Salihu, P. Kumam, S.M. Ibrahim and H.A. Babando, A sufficient descent hybrid conjugate gradient method without line search consideration and application. *Eng. Comput.* **41** (2024).

[41] N. Salihu, P. Kumam, S.M. Ibrahim and W. Kumam, Some combined techniques of spectral conjugate gradient methods with applications to robotic and image restoration models. *Numer. Algor.* **100** (2024) 1–41.

[42] N. Salihu, P. Kumam, I.M. Sulaiman, I. Arzuka and W. Kumam, An efficient Newton-like conjugate gradient method with restart strategy and its application. *Math. Comput. Simul.* **226** (2024) 354–372.

[43] N. Salihu, P. Kumam, I.M. Sulaiman and S. Salihu, A descent matrix-free nonlinear conjugate gradient algorithm for impulse noise removal. *NCAO* **3** (2024) 25–46.

[44] I.M. Sulaiman, M. Malik, A.M. Awwal, P. Kumam, M. Mamat and S. Al-Ahmad, On three-term conjugate gradient method for optimization problems with applications on covid-19 model and robotic motion control. *Adv. Contin. Discret. Model.* **2022** (2022) 1–22.

[45] X. Wu, A new spectral Polak–Ribière–Polak conjugate gradient method. *ScienceAsia* **41** (2015) 345–349.

[46] L. Zhang, W. Zhou and D.-H. Li, A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **26** (2006) 629–640.

[47] L. Zhang, W. Zhou and D. Li, Some descent three-term conjugate gradient methods and their global convergence. *Optim. Method. Softw.* **22** (2007) 697–711.

[48] Y. Zhang, L. He, C. Hu, J. Guo, J. Li and 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.

[49] G. Zoutendijk, Nonlinear programming, computational methods, in Integer and Nonlinear Programming edited by J. Abadie. North-Holland, Amsterdam (1970) 37–86.