Automation of the CFD-based design process for turbomachinery blades using deep reinforcement learning

Innyoung Kim, Jonghyun Chae, Hwijun Bae, and Donghyun You*

Department of Mechanical Engineering, Pohang University of Science and Technology, 77 Cheongam-Ro, Nam-Gu, Pohang, Gyeongbuk 37673, South Korea

*Corresponding author.

Keywords: Automated design optimization; Automatic computational fluid dynamics; Deep reinforcement learning; Turbomachinery

Abstract. A fully automated computational fluid dynamics (CFD)-based design framework for turbomachinery blades is developed using deep reinforcement learning (DRL). Unlike conventional approaches, where the design process, whether manual or automated, must be repeated for each new flow condition, the proposed method employs DRL-based multi-condition (MC) optimization to achieve optimal designs in a single iteration across a range of operating conditions. The framework integrates three key components: (1) Automated Mesh Generation, ensuring optimal-quality grids without manual tuning; (2) Automated CFD Simulations, eliminating repeated simulations for grid convergence tests; and (3) MC Optimization, enabling simultaneous optimization across multiple flow conditions within a unified framework. The efficacy of the proposed methodology is validated for the LS89 turbine cascade across a range of flow conditions including subsonic and transonic regimes. Compared to conventional iterative approaches, which typically require 300 to 500 simulations per flow condition, the present method achieves at least 99.97% of the optimal performance in a single step, eliminating the need for re-optimization across varying flow configurations.

1. Introduction

Designing efficient turbomachinery is essential across a wide range of industries, including aerospace, power generation, and manufacturing. However, the design of turbomachinery blades presents a formidable challenge due to the necessity of accommodating a broad range of flow conditions. In practical applications, aerodynamic conditions vary continuously from the hub to the tip due to blade rotation and spanwise inflow variations. The complexity is further amplified in multi-stage configurations, where aerodynamic interactions between upstream and downstream blades significantly alter local flow characteristics. Conventionally, computational fluid dynamics (CFD) workflows for turbomachinery design, encompassing geometry parameterization, mesh generation, solver setup, and iterative optimization, are conducted separately for each operating condition. Such manual and repetitive processes become increasingly labor-intensive and time-consuming when multiple or continuously varying flow conditions must be considered. Although automating the design process reduces reliance on manual intervention and domain-specific expertise, handling multiple flow conditions remains computationally expensive, as the automated process must still be repeated for each new flow condition.

To address such limitations, the present study aims to achieve two objectives. First, a fully automated design framework is developed to eliminate manual intervention across all stages, from mesh generation to CFD simulation and optimization. Building on prior advancements in deep reinforcement learning (DRL)-based methods for automated mesh generation [1] and non-iterative CFD convergence [2], the current study integrates the prior techniques with a new design optimization method to form a unified pipeline. Second, a multi-condition (MC) optimization framework is developed to handle multiple flow conditions within a single optimization cycle. By incorporating various flow configurations into a unified DRL-based framework, repeated optimizations for changing conditions are eliminated, significantly improving the computational efficiency of the design process.

The paper is organized as follows: Section 2 outlines the methodology, including the governing equations for CFD, the concept of DRL-based MC optimization, and the DRL framework for achieving a fully automated design process by integrating three key components: automated mesh

generation, automated CFD simulations, and MC optimization. Particular emphasis is placed on MC optimization as the novel contribution of this study. Section 3 evaluates the effectiveness of the proposed method for the LS89 turbine cascade under varying flow conditions, followed by concluding remarks in Section 4.

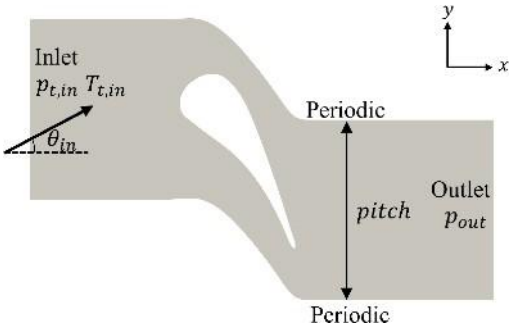

Figure 1: Computational configuration for simulating flow around a turbomachinery blade.

2. Methodology

2.1. Computational fluid dynamics

As a foundational step, the present work focuses on optimizing two-dimensional (2D) profile of turbomachinery blades. Since a three-dimensional (3D) blade is fundamentally a stack of optimized 2D profiles, and multi-stage turbomachinery can be viewed as an extension of 3D design, establishing a robust 2D design method is a prerequisite for efficient turbomachinery design. Thus, 2D compressible Reynolds-averaged Navier—Stokes (RANS) equations are solved to simulate fluid flow around turbomachinery blades, expressed as follows:

$$\nabla \cdot \underbrace{\begin{Bmatrix} \rho \boldsymbol{v} \\ \rho \boldsymbol{v} \otimes \boldsymbol{v} + \bar{\bar{I}} p \\ \rho E \boldsymbol{v} + p \boldsymbol{v} \end{Bmatrix}}_{\text{convective flux}} - \nabla \cdot \underbrace{\begin{Bmatrix} 0 \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}} \cdot \boldsymbol{v} + \kappa \nabla T \end{Bmatrix}}_{\text{viscous flux}} = 0, \#(1)$$

where, , , , , , T, and $\bar{\bar{I}}$ denote the density, the velocity, the total energy per unit mass, the pressure, the thermal conductivity, the temperature, and the identity matrix, respectively. The viscous stress tensor $\bar{\bar{\tau}}$ is defined as $\bar{\bar{\tau}} = \mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T) - 2/3 \mu I(\nabla \cdot \boldsymbol{v})$, where the viscosity μ *is* split into a dynamic part μ_d and a turbulent part μ_t following the Boussinesq hypothesis. The Sutherland's law is employed to obtain μ_d, whereas μ_t is obtained using a turbulence closure model. In this work, the shear stress transport–k– (SST-k-) [3] model is utilized, blending the k- formulation with the k- model, ensuring accurate predictions in both near-wall and core-flow regions. The thermal conductivity is given by $\kappa = (c_p \mu) / Pr$, where c_p is the specific heat capacity at constant pressure and is the Prandtl number. For the equation of state, a perfect gas is employed. The governing equations are solved using the open-source software SU2 [4] where a finite volume method (FVM) is employed for the discretization. A centered Jameson–Schmidt–Turkel (JST) scheme [5] is used for the discretization of the convective fluxes, and a weighted least-squares method is used to compute the gradients required to compute the viscous fluxes.

The computational configuration is illustrated in figure 1. The total pressure $p_{t,in}$, the total temperature $T_{t,in}$, and inflow angle θ_{in} are imposed at the inlet boundary. The static pressure p_{out} is specified at the outlet. Periodic conditions are applied along the upper and lower boundaries, separated by the blade spacing ($pitch$). No-slip and adiabatic wall conditions are applied at the blade surface.

2.2. Single-step deep reinforcement learning for multi-condition optimization

DRL is a method to solve complex decision-making problems by learning a policy $\boldsymbol{\pi}(\boldsymbol{s})$, which maps a given state to an action . The goal is to maximize the cumulative reward $\mathcal{R}_t$, defined as: $\mathcal{R}_t = \sum_{i=t}^{T} \lambda^{(i-t)} r(\boldsymbol{s}_i, \boldsymbol{a}_i), \#(2)$ where $_t$ is the reward at step , is the terminal step, and $\lambda \in [0,1)$ is a discount factor. In single-step DRL, decision-making is completed within a single step, and the formulation simplifies to $\mathcal{R} = r(\boldsymbol{s}, \boldsymbol{a})$, making it suitable for optimization tasks, as there are no subsequent problems to address once the optimal solution is discovered.

MC optimization [6] extends the single-step DRL framework by introducing a condition vector within a condition space $\boldsymbol{\phi}$. An MC optimization problem is formulated as: $\max_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{c})$ subject to $\boldsymbol{x} \in \boldsymbol{\Omega}$, $\boldsymbol{c} \in \boldsymbol{\phi}, \#(3)$ where is a decision vector and $\boldsymbol{f}$ is either a single- or multi-objective function. The objective is to identify an optimal decision vector * for all conditions $\boldsymbol{c} \in \boldsymbol{\phi}$. By incorporating the condition vector into the state representation, DRL can learn the correlation between conditions and the actions leading to optimal results across the entire condition space. This approach eliminates the need for re-optimization whenever conditions change, providing a clear advantage for automating CFD-based design processes, where both geometry and flow conditions vary substantially.

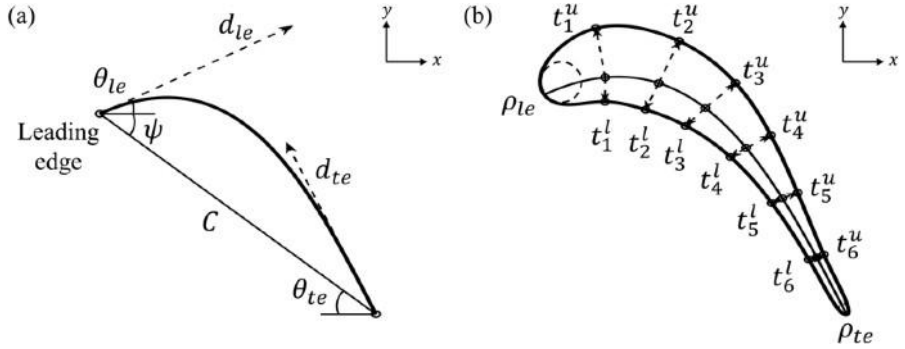

Figure 2: Blade parametrization method. (a) Camber line generation. (b) Blade profile generation.

2.3. Reinforcement learning framework for automating the design process

To achieve a fully automated design process, the proposed framework comprises three DRL-based components: (1) Automated Mesh Generation, which generate meshes with optimal geometric quality, eliminating the need for manual adjustments; (2) Automated CFD Simulations, which produce grid-converged solutions non-iteratively, removing the need for repeated CFD processes for convergence tests; and (3) MC Optimization, which enables blade design across various flow configurations. Once trained, the framework generates optimal blade profiles directly, without requiring re-optimization, even under different operating conditions.

As components (1) and (2) have already been developed and detailed in prior works [1] and [2], respectively, the present study focuses on (3), which constitutes the novel contribution. Building on MC optimization in equation (3), the condition vector is defined as the initial blade profile and the operating condition. The blade profile is parameterized using the method of Agromayor *et al.* [7], with 20 parameters of $(C, \psi, \theta_{le}, \theta_{te}, d_{le}, d_{te}, \rho_{le}, \rho_{te}, t_1^u, \ldots, t_6^u, t_1^l, \ldots, t_6^l)$, as illustrated in figure 2. First, the camber line is constructed using the first six parameters (figure 2a). The chord length is determined from the leading edge at a stagger angle . The camber line is then defined by the metal angles θ_{le} and θ_{te} at the leading and trailing edges, together with tangent proportions d_{le} and d_{te}. Once the camber line is formed, the upper and lower profiles of the blade are generated (figure 2b). The radii ρ_{le} and ρ_{te} define the curvature at the leading and trailing edges, while the thickness distributions are specified by $(t_1^u, \ldots, t_6^u)$ and $(t_1^l, \ldots, t_6^l)$ for each side.

The flow conditions are represented in a nondimensionalized form to enhance the learning efficiency. The boundary conditions $p_{t,in}$, $T_{t,in}$, and p_{out} along with the axial chord length C_{ax} are transformed into the isentropic Reynolds number and Mach number as:

$$Ma_{is,out} = \sqrt{\frac{2}{\gamma - 1}\left[\left(\frac{p_{t,in}}{p_{out}}\right)^{(\gamma-1)/(\gamma)} - 1\right]}, \quad Re_{is,out} = \frac{\rho_{is,out} U_{is,out} C_{ax}}{\mu_{is,out}}, \#(4)$$

where γ is the specific heat ratio, $\rho_{is,out}$ and $\mu_{is,out}$ are computed using the ideal gas law and Sutherland's law, respectively. The isentropic velocity $U_{is,out} = Ma_{is,out}\sqrt{\gamma R T_{is,out}}$ where is the gas constant, and $T_{is,out} = T_{t,in}(p_{out}/p_{t,in})^{(\gamma-1)/(\gamma)}$. The decision vector represents deviations from the initial blade profile. The objective function $f(\boldsymbol{x}, \boldsymbol{c})$ is defined as the static-to-static isentropic efficiency to account for the comprehensive performance measure of the turbine blade, calculated as follows:

$$\eta_{is} = \frac{T_{in} - T_{out}}{T_{in} - T_{in}(p_{out}/p_{in})^{(\gamma-1)/(\gamma)}}, \#(5)$$

where the and are the static temperature and the static pressure, respectively.

The proposed framework for solving MC optimization is illustrated in figure 3. The state consists of the initial blade geometry and the flow condition, which is randomly assigned to an agent. The agent modifies the blade profile based on the executed action. After each modification, the agent automatically generates the optimal mesh and performs a CFD simulation to obtain a grid-converged solution. Once the agent has completed the CFD simulation, the isentropic efficiency is calculated as the reward. The data of states, actions, and rewards are accumulated and used to update the network using the actor-critic algorithm.

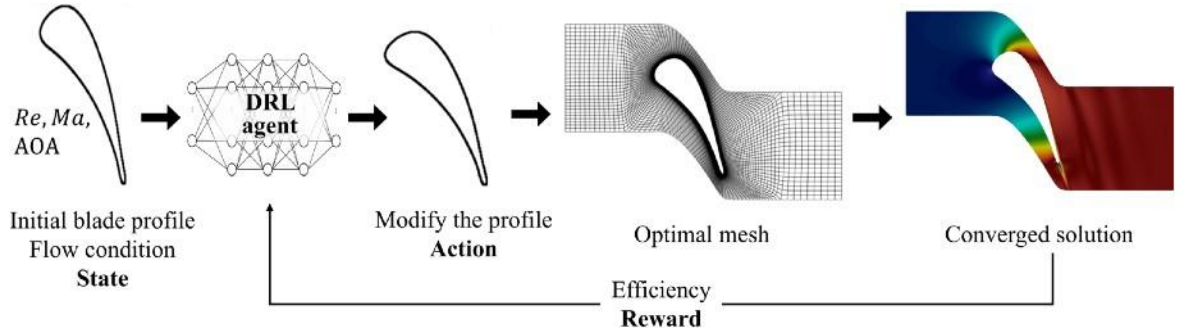

Figure 3: Reinforcement learning framework for automating the design process of turbomachinery blades.

3. Results and discussion

The proposed automated optimization pipeline is examined for the LS89 turbine cascade, a representative high-pressure turbine nozzle guide vane used in modern aeroengines, designed at the von Karman Institute for Fluid Dynamics (VKI). MC optimization is performed across a range of flow conditions as $5.89 \times 10^5 \leq Re_{is,out} \leq 7.82 \times 10^5$ and $0.84 \leq Ma_{is,out} \leq 1.02$, including subsonic and transonic test conditions derived from experimental data [8]. Since the validation of CFD simulation results against experimental data, as well as grid convergence verification, has already been established in prior literature [2], the present work focuses on analyzing the optimization outcomes. Figure 4 illustrates the reward, defined by the isentropic efficiency, as a function of the number of episodes during the optimization process. Statistical convergence is achieved after approximately 3,000 simulations, with improvements of less than 0.01% observed over the preceding 100 episodes.

To validate the efficacy of the proposed optimization framework, the results are compared with conventional iterative optimization for two cases: MUR43 ($Re_{is,out} = 5.9 \times 10^5$ and $Ma_{is,out} = 0.84$) and MUR47 ($Re_{is,out} = 7.77 \times 10^5$ and $Ma_{is,out} = 1.02$), each serving as a benchmark condition for subsonic (MUR43) and transonic (MUR47) flows based on the experimental study [8]. For a fair comparison, conventional optimization employs the same algorithm as the MC

approach but fixes the flow condition during training, necessitating separate optimization runs for each operating point. To minimize stochastic effects, each test condition is evaluated five times with different random seeds. The objectives of the comparative study are: (1) to verify whether the isentropic efficiency achieved in a single iteration by the proposed method matches the optimal solutions derived from the conventional approach, and (2) to quantify computational efficiency by comparing the number of iterations required for convergence.

As shown in figure 5, the present framework achieves comparable results in a single iteration, differing by only 0.001% and 0.027% from the conventional approach for MUR43 and MUR47, respectively. However, the conventional method requires 325 and 543 iterations for MUR43 and MUR47, respectively, to reach the same level of performance achieved by the present approach. Crucially, while the conventional iterative method necessitates re-optimization for every new condition, the proposed framework generates near-optimal designs across the entire condition space in a single step. This eliminates the need for repetitive CFD simulations whenever operational conditions change, offering significant computational efficiency in turbomachinery design.

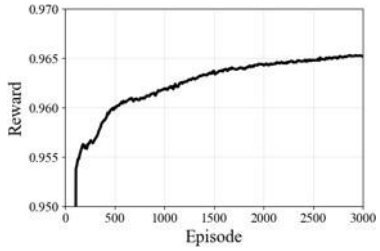

Figure 4: Reward as a function of the episode.

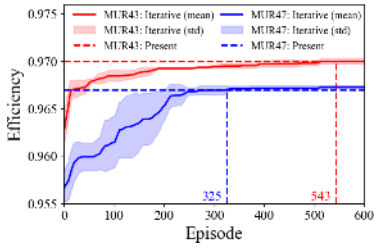

Figure 5: Efficiency as a function of the episode, comparing the performance of conventional (iterative) and proposed optimization methods.

Figure 6 presents contour plots of the Mach number for the baseline and optimized blade profiles. For MUR43, the optimized profile exhibits increased curvature, inducing greater flow acceleration relative to the baseline geometry. Consequently, the Mach number along the suction side locally exceeds unity. However, no significant discontinuity or strong shock wave is generated. Instead, careful adjustments to the profile allow the optimized design to enhance acceleration while mitigating the losses typically associated with shock formation, leading to a 0.812% improvement in isentropic efficiency. For MUR47, the prescribed boundary condition of $Ma_{is,out}$ exceeds unity, making a shock wave inevitable. Nonetheless, the DRL agent delicately modifies the blade geometry to reorient the shock wave, preventing it from propagating to the outlet boundary. This confinement reduces the adverse effects of abrupt compression on the downstream flow and yields a 1.20% gain in isentropic efficiency.

4.Concluding remarks

An automated computational fluid dynamics (CFD)-based design framework for turbomachinery blades has been developed using deep reinforcement learning (DRL). The framework integrates three key components into a unified pipeline: (1) Automated Mesh Generation, (2) Automated CFD Simulations, and (3) Multi-Condition (MC) Optimization. The proposed approach not only automates mesh generation and flow analysis but also eliminates the need for repeated re-optimization under varying flow conditions, leveraging the capability of DRL to learn the correlation between conditions and optimal actions across the entire condition space. The effectiveness of the proposed method has been demonstrated using the LS89 turbine cascade across a range of flow conditions, including subsonic and transonic regimes. Compared to conventional iterative approaches, which typically require 300 to 500 simulations per specific flow condition, the present method

achieves 99.97% of the optimal performance in a single step across a defined range of operating conditions.

Looking ahead, a natural extension of the present framework is the optimization of three-dimensional blades. By incorporating spanwise flow variations where aerodynamic conditions such as the angle of attack vary continuously from hub to tip due to blade rotation, the present method can optimize stacked blade sections without requiring independent optimization at each spanwise location. Furthermore, extending the framework to multi-stage turbomachinery is a promising next step, as upstream and downstream interactions dynamically modify flow conditions across stages. The MC optimization approach is particularly well suited for such applications, as it inherently accounts for variations in operating conditions, eliminating the need for iterative re-optimization at each stage.

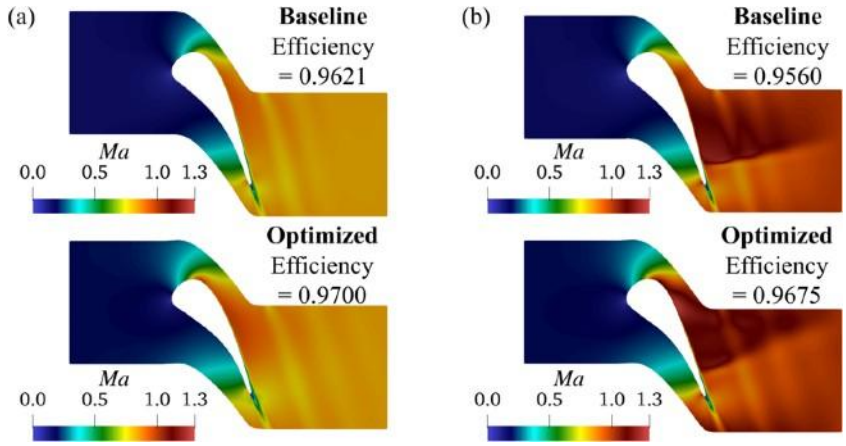

Figure 6: Contour plots of the Mach number for the baseline and optimized blade profiles: (a) MUR43 and (b) MUR47.

Acknowledgements

The work was supported by the National Research Foundation of Korea (NRF) under the Grant Number NRF-2021R1A2C2092146 and RS-2023-00282764 and the Korea Institute of Energy Technology Evaluation and Planning (KETEP) under Grant Number RS-2023-00242282.

References

[1] I. Kim, S. Kim, D. You 'Non-iterative generation of an optimal mesh for a blade passage using deep reinforcement learning', *Computer Physics Communications*, 294 (2024) 108962. https://doi.org/10.1016/j.cpc.2023.108962

[2] I. Kim, J. Chae, D. You 'Optimal mesh generation for a non-iterative grid-converged solution of flow through a blade passage using deep reinforcement learning', *arXiv preprint*, arXiv:2402.15079, 2024.

[3] F. R. Menter 'Two-equation eddy-viscosity turbulence models for engineering applications', *AIAA Journal*, 32 (8) (1994) 1598–1605. https://doi.org/10.2514/3.12149

[4] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, J. J. Alonso 'SU2: An open-source suite for multiphysics simulation and design', *AIAA Journal*, 54 (3) (2016) 828–846. https://doi.org/10.2514/1.J053813

[5] A. Jameson, W. Schmidt, E. Turkel 'Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes', *14th Fluid and Plasma Dynamics Conference*, 1981, p. 1259. https://doi.org/10.2514/6.1981-1259

[6] S. Kim, I. Kim, D. You 'Multi-condition multi-objective optimization using deep reinforcement learning', *Journal of Computational Physics*, 462 (2022) 111263. https://doi.org/10.1016/j.jcp.2022.111263

[7] R. Agromayor, N. Anand, J.-D. Müller, M. Pini, L. O. Nord 'A unified geometry parametrization method for turbomachinery blades', *Computer-Aided Design*, 133 (2021) 102987. https://doi.org/10.1016/j.cad.2020.102987

[8] T. Arts, M. Lambert de Rouvroit, A. Rutherford 'Aero-thermal investigation of a highly loaded transonic linear turbine guide vane cascade. A test case for inviscid and viscous flow computations', *NASA STI/Recon Technical Report N*, 91 (1990) 23437.