The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

MULTIDISCIPLINARY MODULAR APPROACH TO KINEMATIC MECHANISM SYNTHESIS

NASTARAN SHAHMANSOURI[1], HYUNMIN CHEONG[1], ALEXANDER TESSIER[1] AND ADRIAN BUTSCHER[1]

[1] Autodesk Research,
661 University Ave, Toronto, ON, M5G 1M1, Canada
e-mail: firstname.lastname@autodesk.com

Key words: Multidisciplinary Design Optimization, Mechanism Synthesis, Component-based Modelling, Gradient-based Optimization, Direct Collocation

Summary. In kinematic mechanism synthesis, the goal is to find the optimal configuration and parameters of a mechanism system that produces desired mechanical performance such as motion or force. For a problem involving a complex set of requirements, the optimal system often comprises of many mechanism components, known as Mechanical Building Blocks (MBBs). For example, a complex power transmission system is created with a series of gears, shafts, belts, etc.

During the search for an optimal system, the algorithm must be able to evaluate the performance of a candidate system made up of an arbitrary collection of building blocks. To address this challenge, we propose modular modelling of the MBBs that can be composed on-the-fly as a system of equations to be solved. This approach is largely based on multidisciplinary design optimization framework, where the model is composed by considering all relevant disciplines simultaneously to find an optimal solution.

In this work, we present the first set of MBBs modelled so far, and three use cases where these building blocks are automatically composed to create a complex mechanism system and analyzed to find the optimal parameters of the system. Our approach is implemented using Dymos, which employs modular analysis and unified derivatives (MAUD) for computing the total derivatives out of the partial derivatives of individual building blocks for gradient-based optimization and a direct collocation method for integrating the kinematic equations. In summary, our work demonstrates the value of the multidisciplinary design optimization approach in solving a mechanism synthesis problem.

1 INTRODUCTION

The realm of computational mechanism design and optimization has been extensively researched, leading to diverse strategies and methods for making efficient designs. The assessment of a prospective mechanism design, composed of a diverse assortment of components, necessitates simulation of each component and establishing the interrelationships between different components. This process can be demanding as the design process often requires balancing the contradictory constraints of these components while maintaining the continuity of certain variables throughout the mechanism. Hence, the design and optimization process involves different key ele including a strategy for novel mechanism creation in the conception phase, the ability to model the governing physics of a given design, the examination of geometrical attributes of a design, an apt optimization approach to adjust design parameters. In addition to these aspects, for a fully automated process a workflow governing the iterative process of design modeling and optimization is required.

Varied efforts in the key aforementioned aspects of mechanism design and optimization highlight the dynamic nature of the field. Within the context of conceptual mechanism design, Kota and Chiou introduced the concept of kinematic building blocks in the conceptual design of mechanisms, thus enhancing the design process with a more structured approach [1, 2]. Han et al. presented a novel approach to mechanical conceptual design, utilizing graph theory and polynomial operation for computational synthesis [3]. Shea's work on the application of spatial grammar for the computational design synthesis of virtual soft robots, proposes an innovative path to design automation and optimization [4]. Cagan et al. offered a comprehensive framework for computational design synthesis, laying the groundwork for a systematic and efficient automated synthesis process [5]. In more recent work, Bayat et al. conducted research on the generation, modeling, and optimization of single-type multi-component systems, and introduced strategies for information extraction from optimized designs and discrete-domain configuration optimizations using machine learning techniques [6, 7, 8].

During the search for an optimal mechanism design using an automated approach, the workflow of algorithms must be able to evaluate the performance of a candidate system made up of an arbitrary collection of building blocks with different-types. One approach to tackling the challenge of evaluating each design for effective mechanism design, optimization, and search is to integrate modular and parametric capabilities within the design process. This integration can facilitate a comprehensive design optimization process capable of generating diverse design alternatives while ensuring the optimized performance of each design. Such a modular approach towards mechanism design and optimization provides considerable benefits for design generation, search, and optimization, fostering a systematic design process. It also extends the design lifecycle, enabling easy modifications and upgradeability. Compared to alternative methods, this approach could result in significant time and cost savings by allowing changes to individual modules without affecting the overall system.

This work is an attempt to address some of the challenges related to modular-based design and optimization of mechanism. Specifically, we first discuss the steps for a systematic workflow for efficient modeling and parametric optimization of mechanisms, aiming the early design stages. This workflow focuses on three primary aspects: 1) Modularity: using MBBs to create new mechanisms, 2) Generalizability: the possibility of covering diverse types of physical or geometrical attributes, introducing new MBBs, off-the-shelf components, and equations that define problem specifications or MBB characteristics, and 3) Optimality: the compatibility with a gradient-based optimization frameworks.

Next, we introduce the design and optimization methodology, which instantaneously formulates mechanisms as a system of equations. This methodology draws heavily from the multidisciplinary design optimization framework, wherein an optimal solution is sought by concurrently considering all relevant disciplines. We utilize MAUD to calculate total derivatives from the partial derivatives of individual MBBs for gradient-based optimization. To substantiate our proposed framework, we exhibit the initial set of MBBs modeled, three instances where MBBs are composed to form three mechanisms, subsequently these mechanisms are analyzed for optimal parameter determination.

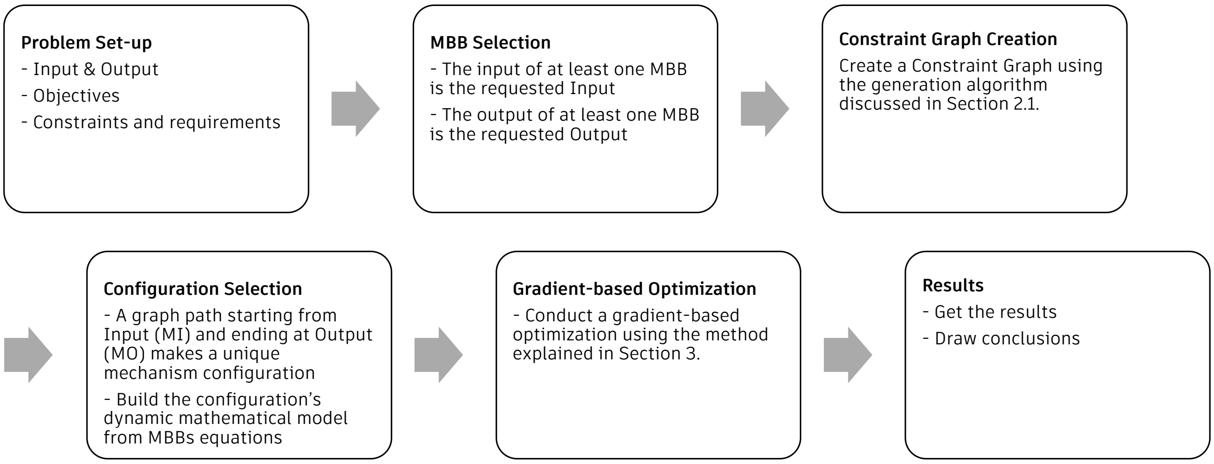


Figure 1: The proposed mechanism generation and optimization workflow.

2 Proposed Framework

2.1 Workflow

Figure 1 proposes a workflow supporting making mechanisms from modular units, simulating them, and conducting parametric optimization on them. The main idea is to compose a chain of compatible MBBs to form a mechanism. The modular units (MBBs) can cover various mechanical building blocks which are widely used in industry for making different mechanisms. The workflow utilizes a constraint graph algorithm generation procedure, described as follows: MBBs form nodes and directed edges' connects two nodes where the output of the first node matches the input of the second one. To create the graph, first two special nodes called MI and MO nodes, respectively containing the Input and Output information of the mechanism, are made. Next, starting from MI, the Directed edges' rule described above is used to connect MI to compatible MBBs. Then, the previous step is continued in a recursive manner by connecting an MBB node to subsequent compatible MBBs, forming new nodes. If the output of a MBB node is compatible with MO, the node can be connected to MO. The procedure can stop anytime when at least 1 edge connects to MO. Refer to [3] for more detailed.

2.2 Parametric Optimization

This work incorporates the concepts of MDO, MAUD, collocation, and phase approaches to provide a robust and efficient framework for the analysis and optimization of the mechanisms composed by a chain of compatible MBBs. The optimization framework supports gradient-based optimizations which can be very fast and efficient in evaluating different parameter options. In comparison to the other optimization approaches, such as population based or heuristic approaches, this is the only method which can guarantee a global optimum for a problem. The MDO approach effectively interlinks different disciplines and domains, allowing for a comprehensive optimization process. In the context of MAUD, each module or discipline, represented by an individual component or group, can independently calculate its outputs and partial derivatives. These are then unified into a larger, interconnected system that seamlessly integrates these modules, enabling a holistic optimization. The unified derivatives concept allows the computation of gradients across the full system, enabling gradient-based optimization. This method is efficient and accurate as it eliminates the need for separate gradient calculations for each module, thereby reducing computational errors and enhancing the overall optimization process. The collocation approach involves approximating a system's responses at several points (collocation points) within a design space. By analyzing these points, the system's behavior can be predicted across the entire design space, enhancing the model's accuracy and efficiency. Here, we use OpenMDAO and Dymos to implement these concepts. The collocation approach is a vital facet of OpenMDAO's MDO process [9, 10, 11, 12].

This work also incorporates the concept of phase in OpenMDAO and Dymos, where a phase can be thought of as a significant stage or segment in the mechanism's operation or process. It is essentially a defined period in the mechanism simulation where specific conditions or rules apply. One of the key features of phases is their ability to exist as independent groups within a model, but they can also be linked together to maintain continuity among certain variables across phases. Each phase can have its own set of equations of motion, and conditions. Phases also play a crucial role in defining constraints in the Dymos, where boundary constraints can be set on variable values at the start or end of a phase. This flexibility allows for a more accurate and tailored optimization process, catering to the unique requirements of each phase.

3 Methodology

3.1 MBBs: Examples and the Derivatives

Table 1 presents a set of 9 MBBs widely used in mechanism designs. The table includes an illustration of the MBBs with their geometrical and physical attributes, equation to obtain the position, velocity, acceleration (where applicable), and the force/torque of the MBBs. In this table, $_i$ and $_o$ denote input and output variables or parameters, respectively. Note that this table is not meant to present an exhaustive list of MBBs. It provides a list of some fundamental MBBs, a number of which are employed in the case-studies of this work. This list is expandable.

3.2 Optimization Approach

Following the workflow presented in Section 2.1, the compatible MBBs can be merged to make a specific mechanism with a particular goal. Given the governing equation of the MBBs, the governing equations of the forming mechanism can be created. These equations enforce the dynamics of a system and make the modeling and simulation of a desired mechanism possible. Considering the design requirements, the objective function and constraints of an optimization are defined. Following Section 2.2, the optimization step tweak the design variables of the mechanism to ensure optimal performance of a system given its desired objective. In this research, we present 3 cases studies where mechanisms are build, modeled and optimized using the aforementioned methods. Note that all case-studies are solved using Dymos' pyOptSparse [13] implementation of the IPOPT optimizer [14]. We leveraged the concept of phase in the third case-study, where the design and optimization of a multi-functionality mechanism is investigates and the problem is broken down into four phases to handle different stages of the trajectory, refer to Sections 4.3 and 5.3

Table 1: Mechanical Building Blocks

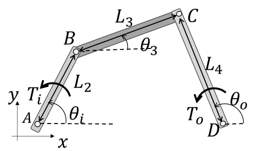
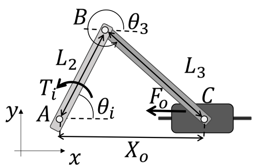
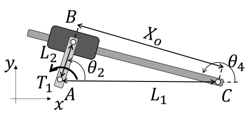
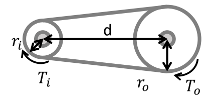
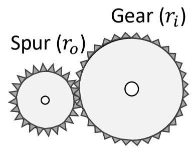
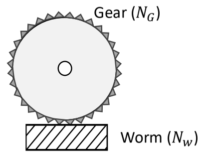
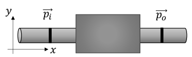
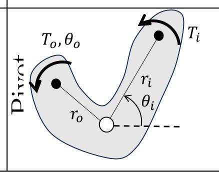
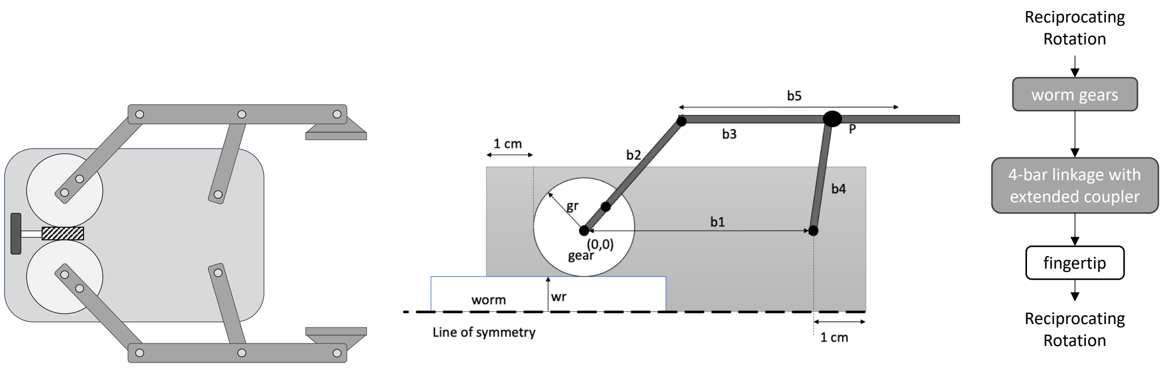
	Image	Equation
Four-bar Linkage		$\begin{bmatrix} L_3\cos\theta_3 - L_4\cos\theta_o = L_1 - L_2\cos\theta_i \\ L_3\sin\theta_3 - L_4\sin\theta_o = -L_2\sin\theta_i \end{bmatrix}$ $\begin{bmatrix} -L_3\sin\theta_3 & L_4\sin\theta_o \\ L_3\cos\theta_o & -L_4\cos\theta_o \end{bmatrix} \begin{bmatrix} \dot{\theta_3} \\ \dot{\theta_o} \end{bmatrix} = \begin{bmatrix} L_2\sin\theta_i\dot{\theta_i} \\ -L_2\cos\theta_i\dot{\theta_i} \end{bmatrix}$, $\begin{bmatrix} -L_3\sin\theta_3 & L_4\sin\theta_o \\ L_4\cos\theta_o & -L_4\cos\theta_o \end{bmatrix} \begin{bmatrix} \ddot{\theta_3} \\ \ddot{\theta_o} \end{bmatrix}$ $= \begin{bmatrix} L_2(\sin\theta_i\ddot{\theta_i} + \cos\theta_i\dot{\theta_i}^2) + L_3\cos\theta_3\dot{\theta_3}^2 - L_4\cos\theta_o\dot{\theta_o}^2 \\ -L_2(\cos\theta_i\ddot{\theta_i} - \sin\theta_i\dot{\theta_i}^2) + L_3\sin\theta_3\dot{\theta_3}^2 - L_4\sin\theta_o\dot{\theta_o}^2 \end{bmatrix}$ $T_o = T_i L_4/L_2\cos(\pi - \theta_i + \theta_3)\cos\theta_o - \theta_3$
Slider-Rocker-1		$\sin\theta_3 = -L_2/L_3\sin\theta_i$ $X_o = L_2\cos\theta_i + L_3\cos\theta_3$ $\begin{bmatrix} -L3\sin\theta_3 & -1 \\ L3\cos\theta_3 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta_3} \\ \dot{X_o} \end{bmatrix} = \begin{bmatrix} L2\sin\theta_i\dot{\theta_i} \\ -L2\cos\theta_i\dot{\theta_i} \end{bmatrix}$ $\begin{bmatrix} -L3\sin\theta_3 & -1 \\ L3\cos\theta_3 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta_3} \\ \ddot{X_o} \end{bmatrix} = \begin{bmatrix} L2(\sin\theta_i\ddot{\theta_i} + \cos\theta_i\dot{\theta_i}^2) + L_3\cos\theta_3\dot{\theta_3}^2 \\ -L2(\cos\theta_i\ddot{\theta_i} - \sin\theta_i\dot{\theta_i}^2) + L_3\sin\theta_3\dot{\theta_3}^2 \end{bmatrix}$ $F_o = T_i/L_2\cos\theta_3 - \theta_i - \pi/2\cos\pi - \theta_3$
Slider-Rocker 2		$\begin{bmatrix} \tan\theta_4 = fracL_2\sin\theta_2 L_1 - L_2\cos\theta_i \\ X_o = frac{-}L_2\sin\theta_i\sin\theta_4 \end{bmatrix}$ $\begin{bmatrix} X_o\sin\theta_4 & -1 \\ -X_o\cos\theta_4 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta_4} \\ \dot{X_o} \end{bmatrix} = \begin{bmatrix} L_2\sin\theta_i\dot{\theta_i} \\ -L_2\cos\theta_i\dot{\theta_i} \end{bmatrix}$ $\begin{bmatrix} X_o\sin\theta_4 & -1 \\ -X_o\cos\theta_4 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta_4} \\ \ddot{X_o} \end{bmatrix}$ $= \begin{bmatrix} L_2(\sin\theta_i\ddot{\theta_i} + cos\theta_i\dot{\theta_i}^2) - X_o\cos\theta_4\dot{\theta_4}^2 - 2\dot{X_o}\sin\theta_4\dot{\theta_4} \\ -L_2(\cos\theta_i\ddot{\theta_i} - sin\theta_i\dot{\theta_i}^2) - X_o\sin\theta_4\dot{\theta_4}^2 + 2\dot{X_o}\cos\theta_4\dot{\theta_4} \end{bmatrix}$ $F_o = T_i/L_2\cos\theta_4 - \theta_i - \pi/2$

Table 1 Continued.

	Image	Equation
Pulley Belt		$\{\theta_o, \dot{\theta_o}, \vec{T_o}\} =$ $\{\frac{\theta_i}{RR}, \frac{\dot{\theta_i}}{RR}, \vec{T_i}.RR\}$, where $RR = \frac{r_o}{r_i}$
Spur Gear		$\{\theta_o, \dot{\theta_o}, \vec{T_o}\} =$ $\{\frac{\theta}{GR}, \frac{\dot{\theta}}{GR}, \vec{T_i}.GR\}$, where $GR = \frac{r_o}{r_i} = \frac{N_o}{N_i}$
Worm Gear		$\{\theta_o, \dot{\theta_o}, \vec{T_o}\} =$ $\{\frac{\theta_i}{GR}, \frac{\dot{\theta_i}}{GR}, \vec{T_i}.GR\}$, where $GR = \frac{r_o}{l_i}$
Shaft		$\{\theta_o, \dot{\theta_o}, \vec{T_o}\} = \{\theta_i, \dot{\theta_i}, \vec{T_i}\}$
Slider		$\dot{\vec{x}}_o = \dot{\vec{x}}_i$, $\dot{\vec{y}}_o = 0$, $\dot{\vec{z}}_o = 0$, $\vec{F_o} = 0$
L-linkage		$\{\theta_o, \dot{\theta_o}, \vec{T_o}\} = \{\theta_i, \dot{\theta_i}, \vec{T_i}\}$



(a) (b) (c)

Figure 2: (a) Gripper mechanism, (b) gripper modeling , and (c) its MBBs.

4 Case Studies

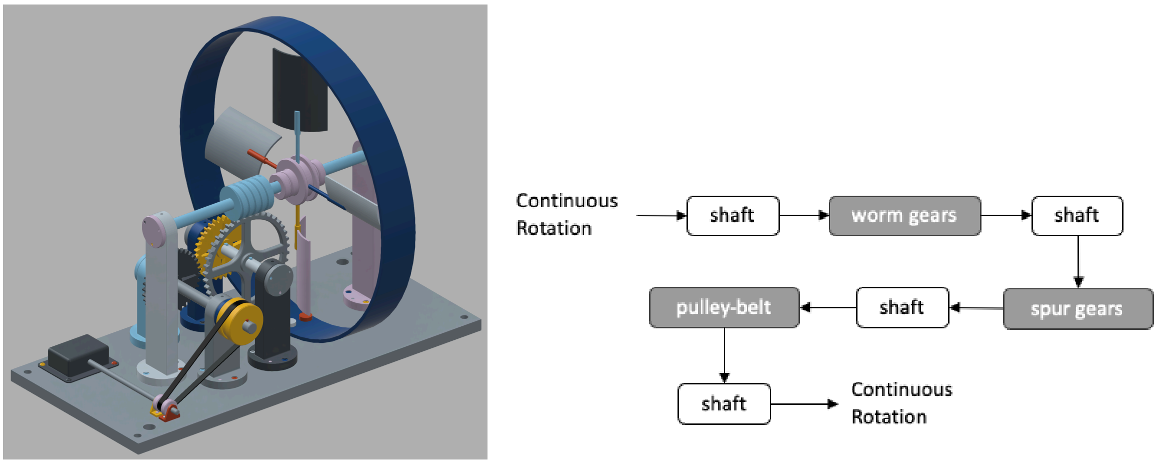
The three case-studies explored in this work are carefully chosen to reflect different level of complexity, cover multiple types of MBBs in mechanisms designs, and represent a multi-functionality mechanism design, while representing real-world or close to real-world scenarios.

4.1 Case-study 1: Gripper

Case-study 1 is about a gripper mechanism illustrated in Figure 2. Grippers are widely used in various tasks and in different industries. Figure 2(a) illustrates an schematic of the studied gripper mechanism and its components. Figure 2(b) depicts the geometrical variables of the mechanism and its components. Here, the line of symmetry shows how half of the model is sufficient to conduct the analysis. Finally, Figure 2(c) shows the chain of MBBs used in modeling this gripper mechanism. The gripper mechanism, constructed of a worm-gear and a four-bar linkage MBBs, is designed to grasp an object based on its clearances and position. The optimization objective of the gripper is to minimize the time taken to grasp the object. The design parameters include bar lengths, i.e. L_2, L_3, L_4, and L_5. The gripper has to be able to lift an object with a minimum weight of $10kg$ and its own maximum weight should not exceed $0.25kg$. The finger-tip needs to travel at most 0.01 m in the x-direction and is at $y = 0.045m$.

4.2 Case-study 2: Wind-turbine

Case-study 2 is about a wind-turbine mechanism illustrated in Figure 3 (a). Figure 3(b) shows the chain of 7 MBBs, including 4 different types, used in the modeling of this wind-turbine mechanism. Here, the goal is to transfer and amplify the angular speed generated by the rotor to the induction motor at another location. To consider the location transfer requirement, we added geometric attributes and translation equations to each MBB. In all translation equations, $\vec{p_i}$ and $\vec{p_o}$ are respectively the locations where the input and output speeds occur, and $\hat{T}$ represents a unit vector that specifies the direction in which the translation occurs. For a shaft with length l, the translation equation is $\vec{p_o} = \vec{p_i} + l\hat{T}$. For a spur gear with radii r_i and r_o, the translation is governed by $\vec{p_o} = \vec{p_i} + (r_i + r_o)\hat{T}$. For a worm-gear with worm radius of r_{worm} and gear radius of r_{gear}, the translation is $\vec{p_o} = \vec{p_i} + (r_{gear} + r_{worm})\hat{T}$. For a pulley-belt with length l,



(a) (b)

Figure 3: (a) Wind-turbine, and (c) its MBBs.

the translation equation is $\vec{p_o} = \vec{p_i} + (r_i + r_o + l)\hat{T}$. The optimization problem is then posed as follows

$$\begin{aligned} \min_{x} \quad & f(W(x)) \\ \text{s.t.} \quad & \vec{p_I} - \vec{p_T}(x) - \vec{p_O} = 0 \\ & \frac{\dot{\theta}_O(x)}{\dot{\theta}_I} = \epsilon \end{aligned} \tag{1}$$

where x is the set of design variables based on the geometric attributes of the MBBs, namely

$$x = \{l^{S_1}, l^{S_2}, l^{S_3}, l^{S_4}, r_1^{SG}, r_2^{SG}, l_1^{WG}, r_1^{WG}, r_2^{WG}, l^{PB}, r_1^{PB}, r_2^{PB}\}. \tag{2}$$

Here, superscripts S_i, SG, WG, and PB refer to a shaft, spur gear, worm gear, and pulley-belt, respectively. W computes the overall weight of the mechanism based on x. $\vec{p_I}$ and $\vec{p_O}$ are the input and output speed locations of the whole mechanism given by the designer. $\vec{p_T}$ is determined by computing the translation equations in series for all the MBB's involved, where the output location $\vec{p_o}$ of a preceding MBB is connected to the input location $\vec{p_i}$ of a following MBB, hence dependent on x. Similarly, $\dot{\theta_I}$ and $\dot{\theta_O}$ are the input and output angular speeds of the whole mechanism. $\dot{\theta_I}$ is given by the designer along with the speed ratio ϵ, while $\dot{\theta_O}$ is determined by computing the speed equations in series for all the MBB's involved.

4.3 Case-study 3: Multi-speed Power Transmission

Case-study 3 is the most complex scenario covering a multi-speed power transmission mechanism illustrated in Figure 4. This multi-functional mechanism is made of 7 MBBs composed in a way to accomplish the specific tasks of delivering 4 different speed ratios. This power transmission creates the four speed ratios by pairing different spur-gear MBBs, namely: (1) S^I-SG^1-S-SG^3-S^O (2) S^I-SG^1-S-SG^4-S^O, (3) S^I-SG^2-S-SG^3-S^O, (4) S^I-SG^2-S-SG^4-S^O. Here, S^I, SG^i, S, S^O respectively represent input shaft, i^{th} spur-gear, middle shaft, and output shaft.

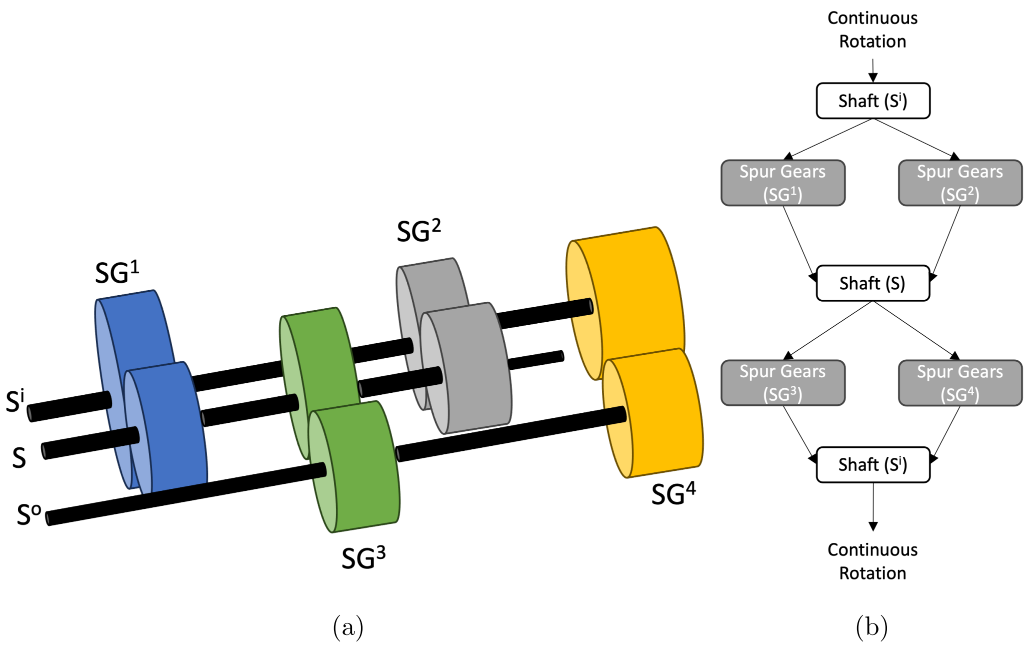


Figure 4: (a) Multi-speed power transmission, (b) Its MBBs.

Figure 4(a) illustrates an schematic of the studied mechanism and its components, i.e. four pairs of spur-gears and three shafts. Figure 4(b) shows a graph representation of the 7 composed MBBs used in modeling this power transmission mechanism. Starting from S^I following a path on this graph gives one MBB pairing strategy. The optimization goal for this case was set to minimize weight subjected to speed and torque constraints. In this scenario, different OpenMDAO *phases* were defined to analyze the whole system. The design variables for this mechanism include the radii of the second gear of SG^1 gear-pair, r_{1-2}, the second gear of SG^2 gear-pair, r_{2-2}, two radii of SG^3 gear-pairs, i.e. r_{3-1}, r_{3-2}, two radii of SG^4 gear-pair, i.e. r_{4-1}, r_{4-2}, totaling 6 variables. Four constraints are imposed on the torque values for four different output torques, T_{1-3}, T_{1-4}, T_{2-3} and T_{2-4}. For the four pair-sets (1 to 4), the minimum output torques are set at $140Nm$, $80Nm$, $20Nm$, and $10Nm$ respectively, while the maximum output torques are set at $160Nm$, $100Nm$, $40Nm$, and $20Nm$ respectively. There are also four constraints on gear radii to ensure fixed locations for shafts 1 and 3. The geometrical constraint, defined by the equation $2 \times r_{1-1} + 2r_{4-2} + r_{4-1} \leq 2 \times r_{1-1}^{init} + 2r_{4-2}^{init} + r_{4-1}^{init}$, ensures that the size of the gearbox does not exceed a specific threshold. This threshold is determined based on the initial value ranges of the variables considered in the problem. Our objective is to minimize the gearbox weight, which is defined as the combined weight of all gears.

5 Results

5.1 Case-study 1: Gripper

In this case, the design parameters, i.e. the bar lengths, can vary by 20% above or below the initial value. The gripper is assumed to be made of Aluminum, with bar cross-sections of $0.0002m^2$. The motor input speed is assumed to be $10rad/s$ and with a motor torque of $75Nm$.

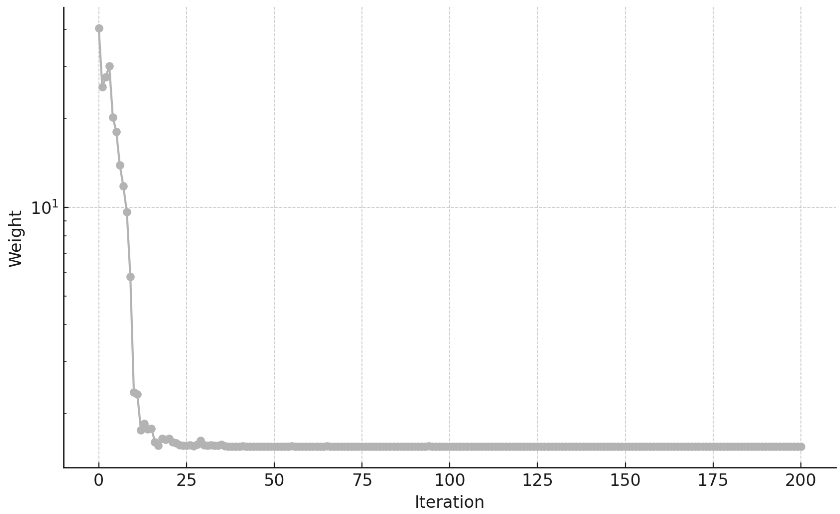


Figure 5: The convergence plot of the objective function for Case-study 2.

In Equations 3, X, Y defines of the gripper finger-tip's location; dX gives the displacement of the finger-tip in the X direction, W gives the gripper's weight, F defines the force exerted on the object by the gripper. Note that in this example a number of equations governing the physical and geometrical attributes of the gripper are defined explicitly to show the capability of defining and incorporating user-defined equations. The optimization resulted in a gripper with a weight of 0.2498 kg and $L2 = 0.0461m, L3 = 0.0195m, L4 = 0.0623m$, and $L5 = 0.0581m$. The optimized gripper demonstrated the capability to lift an object weighing $36.0059kg$, achieving the grasping action in $0.0189seconds$.

$$\begin{aligned}
X &= L_2 \cos\theta_2 + L_5 \cos\theta_3 \\
Y &= L_2 \sin\theta_2 + L_5 \sin\theta_3 \\
dX &= \sqrt{(X_f - X_i)^2} \\
W_{4-bar} &= \rho.A_{bar}.(L_2 + L_4 + L_5) \\
F &= \frac{T}{\sqrt{X_f^2 + Y_f^2}}
\end{aligned} \tag{3}$$

5.2 Case-study 2: Wind-turbine

Here, the optimization goal is to determine the 12 design variables presented in Equation 2 that minimize the weight of the mechanism W, while meeting two constraints related to the output position $\vec{p}_O(x)$ and output speed $\dot{\theta}_O(x)$. Fixed parameter $\vec{p_I}$, $\dot{\theta}_I$, and ϵ are respectively set at $[0.2, 0.05, 0.15]$, 1.0, and 10.0. The bounds for all variables in x are set between $[0.001, 1.0]$. The optimization problem was resolved in 197 iterations, as shown in the convergence plot in Figure 5. The optimal values found for the design variables (2) were $\{0.0404, 0.0346,\ 0.0010, 0.0164, 0.0127, 0.0131, 0.0531,\ 0.0010, 0.0010,\ 0.1804, 0.0058, 0.0067\}$.

5.3 Case-study 3: Multi-speed Power Transmission

Here, the design variables can vary within $+/-20\%$ of the initial value. The first gear radii of SG^1 and SG^2, i.e. r_{1-1} and r_{2-1}, are both set at $0.01m$. The input torque and speed of the motor are $60Nm$ and $10rad/s$, respectively. The mechanism is made of Aluminum and the gear thickness is $t = 0.02m$. The resulting optimal design variable values are $r_{1-2} = 0.0150m$, $r_{2-2} = 0.0027m$, $r_{3-1} = 0.0106m$, $r_{3-2} = 0.0187m$, $r_{4-1} = 0.0208m$, $r_{4-2} = 0.0229m$. The four output torque values are $T_{1-3} = 159.7804Nm$, $T_{1-4} = 99.7000Nm$, $T_{2-3} = 28.7725Nm$ and $T_{2-4} = 16.2417Nm$. The optimized mechanism's weight is $W = 8.5730kg$.

6 Discussion

This study presents a workflow for modeling and optimization of mechanisms using a modular approach. A set of MBBs are presented which can be reused and easily combined to make new mechanisms for different applications and then to simulate and analyze them. The process of making mechanisms by the modular approach of connecting a set of compatible MBBs is presented and demonstrated through three case-studies with multiple levels of complexity. This modular approach allows us to easily add equations of interest for a specific module and apply various constraints on a simulation or optimization problem. The first case-study is a gripper composed of a worm-gear and a four-bar linkage MBBs. Equations are used to define a number of governing equations and constraints of the problem to demonstrate the aforementioned capability. The second case-study demonstrates a more complex mechanism example which is a wind-turbine composed on multiple sets of MBBs, i.e. 7 MBBs with 4 different types. The third case-study, with multi-functional capabilities and a total number of 7 MBBs, is the most complex scenario. The mechanism receives one input and delivers four different outputs.

The workflow also describes a method of conducting parametric optimization on the mechanisms built through the modular process. This capability is also demonstrated through the case studies. For the first case study, the optimization objective was to minimize the time to grasp an object given the clearances and the position of the object. For the second one, the optimization tweaks the parameters of the 7 MBBs to make sure while the output torque requirements are met the wind-turbine meets specific size limitations. The last case is a multi-functional or multi-input - multi-output mechanism. Here, we optimize the parameters of a multi-speed power transmission system and make sure that given one input, four output scenarios are deliverable while meeting the size constraint of the system. To conduct the analysis for this case, four distinct phases were defined to complete the calculations. Each phase models one state of this multi-state scenario. The optimization set-up was done carefully to make sure that the connections and constraints are set-up precisely.

In the future, this work can be expanded in several direction. The MBB dataset can be expanded to include more modules facilitating the modeling and optimization of mechanisms for different applications and with varying levels of complexity. Mechanism modeling and simulation can include further complexity as an example systems with controllers can be modelled as OpenMDAO and Dymos support these capabilities, refer to [6, 7, 8]. Finally, this framework can be integrated as part of a search approache through an automatic generation mechanism strategy. This step can be done by incorporating the constraint graph algorithm method discussed in Section 2.1. The generated and optimized mechanisms can be used inside a search engine working based on a machine learning, population-based, or a heuristic optimization approaches.

7 CONCLUSIONS

This work presents a workflow for modeling and optimization of mechanisms with various constraints using a modular approach. The proposed workflow can facilitate the generation, simulation, and parametric optimization of mechanisms made of different modulus. The modules, or MBBs, can be reused and easily combined to make new mechanisms for different applications. The optimization method of the workflow is a gradient-based optimization which can provide a fast and efficient way of optimizing mechanism designs. This approach is largely based on MDO, where the problem model is composed by considering all relevant disciplines simultaneously to find an optimal solution. Our approach is implemented using Dymos and works based on the concepts of MAUD, direct collocation, and phase in openMDAO and MDO framework.

We presented the first set of MBBs modelled with their physical and geometrical attributes. The process of making mechanisms by the modular approach is demonstrated through three use-cases composed of different modules with various constraints, including a gripper, a wind-turbine, and a multi-speed power transmission mechanism. The case-studies were carefully chosen to reflect different levels of complexity, cover multiple types of MBBs in mechanism designs, and represent a multi-functional mechanism case, while representing real-world or close to real-world scenarios. The optimization procedure is also demonstrated through these case-studies. Each case-study simulates a scenario and optimizes it to find the optimal parameters of its components.

REFERENCES

[1] Kota, S. and Chiou, S.J., 1992. Conceptual design of mechanisms based on computational synthesis and simulation of kinematic building blocks. Research in Engineering Design, 4(2), pp.75-87.

[2] Chiou, S.J. and Sridhar, K., 1999. Automated conceptual design of mechanisms. Mechanism and machine theory, 34(3), pp.467-495.

[3] Han, L., Liu, G., Yang, X. and Han, B., 2020. A computational synthesis approach of mechanical conceptual design based on graph theory and polynomial operation. Chinese Journal of Mechanical Engineering, 33, pp.1-15.

[4] Shea, K., van Diepen, T. (2018). A Spatial Grammar Method for the Computational Design Synthesis of Virtual Soft Robots. ASME DETC conference 2018.

[5] Cagan, J., Campbell, M.I., Finger, S. and Tomiyama, T., 2005. A framework for computational design synthesis: model and applications.

[6] Bayat, S., Shahmansouri, N., Peddada, S.R., Tessier, A., Butscher, A. and Allison, J.T., 2023. Multi-split configuration design for fluid-based thermal management systems. arXiv preprint arXiv:2310.15500.

[7] Bayat, S., Shahmansouri, N., Peddada, S.R., Tessier, A., Butscher, A. and Allison, J.T., 2023. Extracting Design Knowledge from Optimization Data: Enhancing Engineering Design in Fluid Based Thermal Management Systems. arXiv preprint arXiv:2310.16324.

[8] Bayat, S., Shahmansouri, N., Peddada, S.R., Tessier, A., Butscher, A. and Allison, J.T., 2023. Advancing Fluid-Based Thermal Management Systems Design: Leveraging Graph Neural Networks for Graph Regression and Efficient Enumeration Reduction. arXiv preprint arXiv:2311.14874.

[9] OpenMDAO 1.0.3 Alpha documentation. Available at: https://openmdao.readthedocs.io/en/1.0.3/ (Accessed: 3 May 2024).

[10] Gray, J.S., Hwang, J.T., Martins, J.R., Moore, K.T. and Naylor, B.A., 2019. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. Structural and Multidisciplinary Optimization, 59, pp.1075-1104.

[11] Hwang, J.T., 2017. Reconfigurable model execution in the OpenMDAO framework. In 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (p. 3821).

[12] Falck, R., Gray, J.S., Ponnapalli, K. and Wright, T., 2021. dymos: A Python package for optimal control of multidisciplinary systems. Journal of Open Source Software, 6(59), p.2809.

[13] Ella Wu, Gaetan Kenway, Charles A. Mader, John Jasa, and Joaquim R. R. A. Martins, 2020 pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems. Journal of Open Source Software, 5(54), p. 2564.

[14] A. Wächter and L. T. Biegler, "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25-57, 2006.