Mechanics & Industry **26**, 16 (2025)

https://doi.org/10.1051/meca/2024035

Mechanics & Industry

Available online at:
www.mechanics-industry.org

Regular Article

Development of an integrated engineering education model using multi-body closed system modeling, computational analysis, and signal processing analysis

Yonghu Park*

Department of Mechanical Engineering, Sunmoon University, Asan, Chungnam, Republic of Korea

Received: 13 February 2024 / Accepted: 4 December 2024

Abstract. In this study, free high-level programming languages, GNU Octave and Python, are used to model the mechanical part simulating the periodic signal generated in the mechanical system and to link with the signal processing unit that analyzes the generated signal, providing an integrated educational tool capable of recognizing faults in the mechanical system. A design system was developed using GNU Octave to acquire the dynamic characteristics of a closed system in real time, inputting them into the signal processing unit. This is achieved by utilizing the Lagrangian multiplier method to process constraint equations and calculating solutions for differential-algebraic equations using the Runge-Kutta method. Subsequently, a predictive model that simulates real-time cycle data was established, and a design scenario was proposed to assess the stability of the mechanical system by continuously updating and comparing its dynamic characteristic data. In particular, the system structure, where the mechanical part (GNU Octave) and the signal processing part (Python) share data in the form of a .txt file, is designed to improve data processing speed. Additionally, the prediction model, Prophet (developed by Facebook), is incorporated to aid in the development of mechanical products from a hardware perspective, such as simple strength and dynamic characteristics analysis. This is achieved by establishing a user-defined periodic model for unique periodic signals, efficiently processing accumulated data, reviewing reliability based on the difference between the prediction model and actual data, and proposing abnormality detection methods. The system aims to enhance the programming capabilities of mechanical engineering students and industrial personnel by providing hands-on experience with the element technologies used in developing automation systems compatible with electronic equipment and programs.

Keywords: Integrated design system / multi-body dynamics / differential-algebraic equation / Runge-Kutta method / signal process / time series analysis

1 Introduction

Based on various computing technologies, the development of analysis tools that can check the operating mechanisms of machines and equipment beforehand has been continuously conducted, and the application of system control and analysis technologies has been actively pursued with the development of programmable digital technology [1,2]. In the case of general-purpose software companies specializing in design fields such as mechanical design, analysis, and analysis, they are transforming into design tools that non-experts can efficiently utilize, such as transforming into integrated software through technology sharing among related software companies [3]. Regarding technological development status in each area, computational fluid analysis, grid processing technology, solver performance improvement [4], in-shape modeling, standardization of CAD models to increase connectivity with various software [5], and development of specialized design platform technology for the optimization of comprehensive design [6] have been considered. In addition, automation programs are continuously developed for mechanical design based on unique physical phenomena experienced by-products in each manufacturing industry. They are being improved so that engineers can efficiently utilize them in production activities [7–10]. In this way, as engineering tools are improved based on data sharing between machine design, analysis, control technologies, and intelligent element technologies such as machine learning and artificial intelligence, users can be secured by making high-quality engineering results accessible with lower entry barriers [11,12]. However, due to limitations in integrated design tools and the heavy reliance on experts in

* e-mail: yonghuipark@sunmoon.ac.kr

mid-sized industries, there is a pressing need for a cost-effective, integrated design model that can support comprehensive education and training across disciplines. This study addresses these limitations by developing a unified tool that allows undergraduate mechanical engineering students to access integrated, functional education on multi-body dynamics, kinematics, control, and monitoring within a single platform, which is also more accessible to resource-constrained environments [13].

Multi-body dynamics analysis is a study that analyzes the dynamic characteristics of a multi-body system in which several objects are connected. In the field of machinery handling closed chain systems, much attention has been paid to deriving optimal designs through sensitivity analysis between design variables and dynamic characteristics in the system design process, and in the field of aerospace dealing with open chain systems, research on system safety has been the main focus [14]. Research on multi-body dynamics analysis has focused on improving the motion equation derivation method, expanding the scope of analysis, deriving an efficient calculation algorithm according to hardware calculation performance improvement, and improving the software for user convenience. As for multi-body dynamics software, overseas programs like ADAMS, MESA VERDE, and NEWEUL are commercially available, whereas RecurDyn and DAFUL are available in Korea [15]. As such, multi-body dynamics can calculate position, velocity, acceleration, and essential physical quantities, such as the time history of dynamic loads acting on each component, making it an essential discipline for analyzing and designing dynamic characteristics of dynamic systems. For the popularization of dynamics analysis, advanced programs have already developed commercial software that can automatically derive equations of motion and calculate dynamics when a user performs system modeling based on GUI [16]. In addition, the dynamics model derivation method for efficient control in the robot industry. Theoretical studies have been actively carried out, and program development has been promoted a lot [17–20]. In particular, recently, with the development of a MATLAB package program that automatically derives the equations of linear motion of a robot using the TMT method and calculates dynamic characteristics, packages that various users can utilize have been developed [21,22]. In previous studies of this study, an automatic modeling package was developed to analyze the dynamic characteristics of open and closed multi-body systems [23,24]. Using the GNU Octave program and the Symbolic package, the equations of motion and constraint equations of the open-closed multi-degree-of-freedom system were developed, matrix conversion codes were developed, and the codes for deriving numerical solutions were automated using the Newmark and Runge-Kutta methods. The current study builds upon this prior work by providing an educational design case that strengthens comprehension in analysis modeling and response derivation. The developed platform integrates this automation with real-time signal processing to continuously monitor system responses, enabling students to practically apply system engineering, data exchange, and analysis within an educational context.

This research aims to create an integrated analysis system that automates multi-body system analysis modeling and response calculation, simulating real-world mechanical signal generation for students to analyze. The tool links to a signal processing module that examines signal periodicity, supporting students in understanding the fundamentals of signal trend analysis and two-element system dynamics. By providing continuous response evaluation and enabling intelligent mechanical system scenarios, this tool fosters a hands-on, integrative learning experience.

2 Theoretical background

2.1 Dynamical modeling technology of a multi-body mechanical system

The coordinates of the objects constituting an arbitrary mechanical system define the Lagrangian L (Eq. (1)), considering the system's kinetic energy T as an independent variable and the conservative or non-conservative energy V according to the relative motion between each coordinate. For an open system, where no constraint equations are applied, the principle stated in equation (2) holds: among all possible paths when moving from one point to another, the integral of the difference between kinetic and potential energy over time is minimized. After conducting mathematical operations in equation (2) and classifying each term based on the criteria of which physical quantity is multiplied and which motion is related to the term, the general form of the equations of motion (Eqs. (3) and (4)) can be derived in the form of differential equations for an open system without constraints, yielding differential equations based on the relative motion between each coordinate [24].

$$L = T - V = \frac{1}{2}\sum\left(M{\dot{q}_t}^2 + I{\dot{q}_r}^2\right) - \sum U(q_t, q_r), \quad (1)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \frac{d}{dt}\frac{\partial (T-V)}{\partial \dot{q}} - \frac{\partial (T-V)}{\partial q} = Q, \quad (2)$$

$$F_t = [M_t]\ddot{q}_t + [C_t]\dot{q}_t + [K_t]q_t, \quad (3)$$

$$F_r = [M_r]\ddot{q}_r + [C_r]\dot{q}_r + [K_r]q_r, \quad (4)$$

where T is kinetic energy, V is potential energy, q, $\dot{q}$, and $\ddot{q}$ are position, velocity, and acceleration about generalized coordinate, Q is the external force about generalized coordinates, F_t and F_r are the external force for translational motion and torque for rotational motion from Q respectively, q_t, $\dot{q}_t$, and $\ddot{q}_t$ are the translational position, velocity, and acceleration about generalized coordinate from q, $\dot{q}$, and $\ddot{q}$, q_r, $\dot{q}_r$, and $\ddot{q}_r$ are the rotational position, velocity, and acceleration about generalized coordinate from q, $\dot{q}$, and $\ddot{q}$, M and I are the mass and inertia of each object, U is the potential energy of each object, $[M_t]$ and $[M_r]$ are the mass matrix for translational motion and inertia matrix for rotational motion, $[C_t]$ and

$[C_r]$ are the damping matrix for translation and rotational motion respectively, and $[K_t]$ and $[K_r]$ are the stiffness matrix for translation and rotational motion respectively.

In contrast, for a closed system that repeatedly undergoes a particular motion, there is a constraint condition between objects that mutually affects their motion, when calculating the response, equation (5) and the constraint condition are defined, and the Lagrange multiplier method is used to determine the existence of the constraint condition. As a method of finding the extremal value of a multivariate function, it is shown that the extremal value satisfies both the multivariable function and the constraints, and that the tangent of the two functions is parallel [25,26]. When deriving the Lagrangian Equation of motion, equation (7) can be obtained by considering the virtual work done by the virtual displacement δq in the condition where no actual motion occurs, as described in equation (6) for the open system. Based on the condition that the product of the partial derivative of the constraint equation Φ with respect to the virtual displacement and the virtual displacement itself satisfies equation (8), the Lagrangian Equation of motion, incorporating the Lagrange mulitplier theory, is presented as shown in equations (9) and (10). Finally, equation (11) can be derived, representing a differential-algebraic equation in matrix form, and a response that satisfies the constraints can be obtained through numerical analysis.

$$\Phi = 0, \tag{5}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - Q = 0, \tag{6}$$

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - Q\right)\delta q^T = 0, \tag{7}$$

$$\Phi_q \delta q = 0, \tag{8}$$

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - Q\right)\delta q^T - \lambda \Phi_q \delta q = 0, \tag{9}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = Q + {\Phi_q}^T \lambda, \tag{10}$$

$$\begin{bmatrix} [M] & {\Phi_q}^T \\ \Phi_q & 0 \end{bmatrix} \{ q\, \lambda \} = \left\{ \begin{matrix} W \\ -(\Phi_q \dot{q})_q \dot{q} \end{matrix} - 2\Phi_{qt}\dot{q} - \Phi_{tt} \right\}. \tag{11}$$

where Φ is the constraint equation, δq is the virtual displacement with respect to the generalized coordinates, and W is summation of Q and other terms expect the multiplication term of mass and inertia with acceleration, denoted as $[M]q$ from equation (10).

In this study, I focus on mechanical systems that follow holonomic relations, which are typically characterized by position-based constraints as described in kinematics. The main objective is to perform dynamic analysis for systems with holonomic constraints to derive the expected response, and the platform being developed is aimed at detecting anomalies when actual measurements deviate from these expected responses. As such, non-holonomic relations, which involve constraints that depend on velocity or generalized variables, are not considered in our current work. However, non-holonomic systems can be addressed in future research. By transforming the current framework, we could extend the analysis to handle non-holonomic constraints, potentially incorporating additional equations or techniques to represent the velocity-dependent constraints that characterize these systems. To handle non-holonomic relations, I would modify the constraint equations in equation (5) to include velocity dependencies. By organizing these velocity-dependent constraints into matrix form, I can integrate them with the system's dynamics described by the Lagrangian into a system of differential-algebraic equations (DAEs). The system would then be solved numerically using methods like implicit integration or Newmark's method to compute the system's response while respecting the non-holonomic constraints. This would extend the framework to include non-holonomic systems in future work.

2.2 Response derivation technology (numerical integration method)

Various numerical integration methods can be applied to obtain the response by solving the previously defined closed system's differential-algebraic equation. To calculate the response; x of a closed system, the Runge-Kutta method was applied because additional iterative calculations were required to derive an optimal solution that satisfies both the differential equation and the algebraic equation, including the Lagrangian multiplier. For non-stiff equations where responses in closed systems exist within a specific range of time scales, solutions are typically obtained using the ode45 solver [24]. However, in this study, it is necessary to improve the accuracy of the response because the system is modeled in two scenarios: regular operation, where all design factors are in a time-invariant state, and time-varying abnormal operation, where a specific design factor fluctuates in the middle during the mechanical system analysis and modeling process. In this study, the ode78 solver was selected to solve the time-varying algebraic equation. ode78 is an explicit Runge-Kutta method that utilizes 7th-order Fehlberg formulas. The Runge-Kutta method, which increases the accuracy of prediction by dividing the step time h to predict the response at the time $t + h$, which is the next time t, and calculating the amount of change for each division time point It is a method to increase the predictability by applying a coefficient-based higher order term. As the integration method utilizes the seventh and eighth terms in equation (12), a local error of $O(h^9)$ occurs [27].

$$x(t+h) = x(t) + \frac{25}{216}K_1 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5, \tag{12}$$

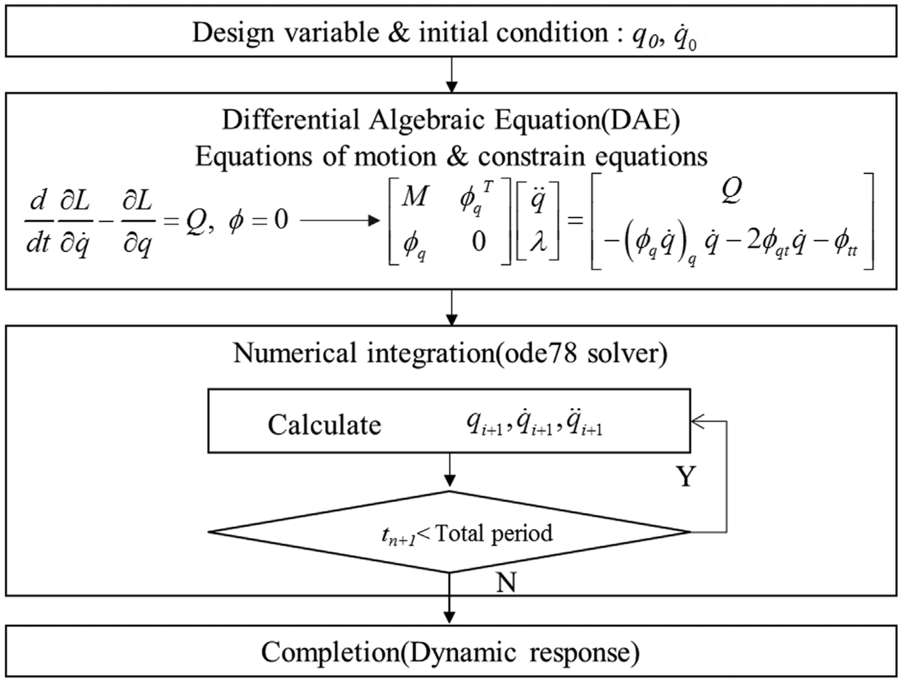

Fig. 1. The flow of the system modeling and numerical integration.

where $K_1 = hf(t, \quad x)$, $K_2 = hf\left(t + \frac{1}{4}h, x + \frac{1}{4}K_1\right)$, $K_3 = hf\left(t + \frac{3}{8}h, x + \frac{3}{32}K_1 + \frac{9}{32}K_2\right)$,- $K_4 = hf\left(t + \frac{12}{13}h, x + \frac{1932}{2197}K_1 - \frac{7200}{2197}K_2 + \frac{7296}{2197}K_3\right)$, and $K_5 = hf\left(t + h, x + \frac{439}{216}K_1 - 8K_2 + \frac{3680}{513}K_3 - \frac{845}{4104}K_4\right)$

Applying the numerical integration method is not a big problem because the system equation, formalized in the form of a matrix from the system model of Figure 1 through the response calculation procedure, is applied to the solver and calculated. It is essential to define the number of degrees of freedom and the coordinate system necessary to express the system response of a closed system, derive the equations of motion and constraint equations, and express them as matrices. This is very helpful in understanding the analysis process.

2.3 Response analysis technology

Regarding the existing response and signal processing analysis, the primary focus has been on detecting specific signals through time domain analysis, statistical methods, frequency domain analysis, etc., of the response obtained from the equation of motion [28]. In particular, when using MATLAB and GNU Octave, system modeling, response derivation, and analysis are performed in one code, and the main focus is on providing technical education in a way that helps learners grasp the physical meaning. However, considering the process of measuring the response occurring in actual hardware, converting it into data, reviewing this data by the analysis unit, and providing information to the user, it is necessary for response analysis to form an independent system for analyzing the hardware signal. In particular, response analysis technologies that have been favored in the past, such as FFT (Fast Fourier Transform), have limitations in analyzing signals with complex periodic and trend trends [29].

This study analyzed response signals using the Prophet model, developed by Facebook, among various response technologies. Prophet has a function that can consider the trend, periodicity, and weekly seasonality of time series data, making it highly effective in analyzing data in human life areas such as stocks and visitor patterns [30]. As the system's degree of freedom increases, various signals can be generated. Therefore, the automatic anomaly detection function, utilizing the latest response analysis technology, was used as a foundational technology for the design.

3 Integrated design system modeling and implementation

3.1 Integrated design system structure

To promote engineering learning for the entire cycle of mechanical system design, manufacture, operation, measurement, and management, a system comprising three interconnected components—a Mechanical Part, a Data Part, and an Analysis Part—was developed (Fig. 2). In the Mechanical Part, it is a subsystem where users can experience the business areas of design, manufacture, and operation by performing closed system multi-body system modeling and response calculation in the GNU Octave space. The Data Part is a subsystem that converts GNU Octave responses into files for engineering activities using responses generated from the mechanical system.

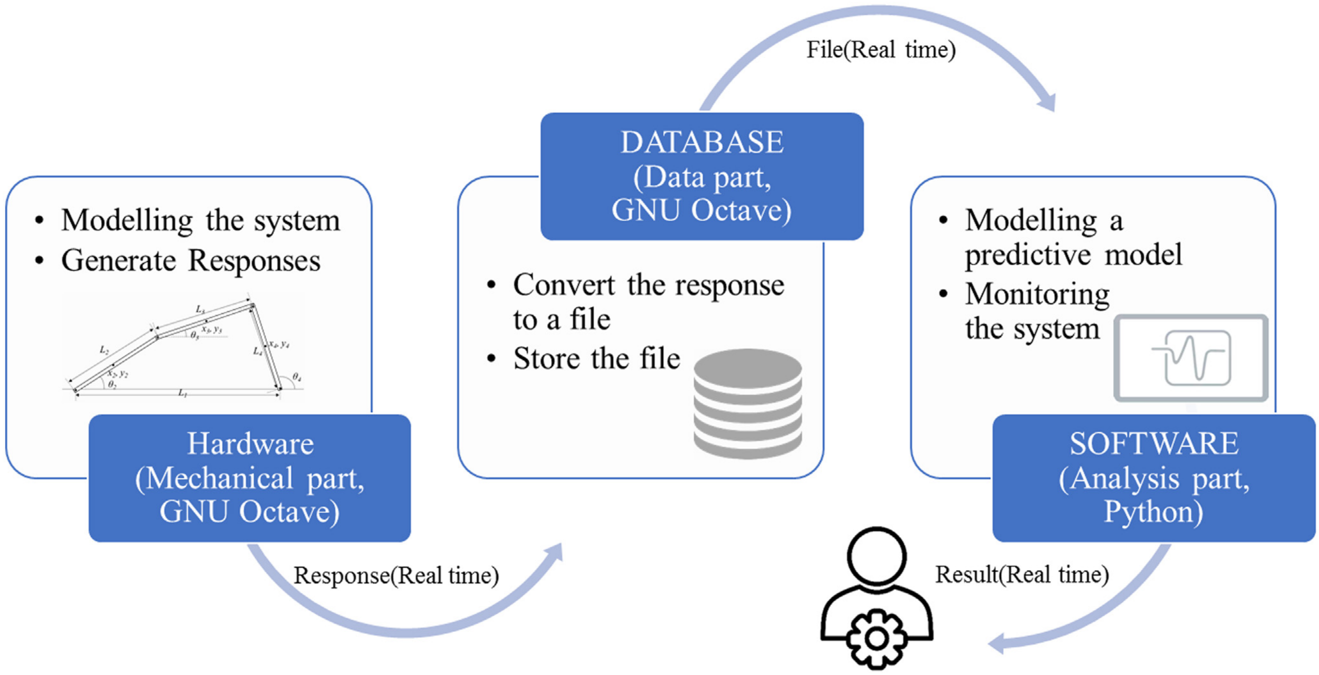

Fig. 2. The structure of the integrated engineering system.

The Analysis Part is a subsystem that establishes a predictive model for response data in the Python space using the acquired files, calculates the difference with real-time data, and reports the system status to the user in real time. By implementing data exchange between the three subsystems, I aim to understand the system structure of machine learning and artificial intelligence-based machine system intelligence.

3.2 Mechanical part

Deriving the response in the actual mechanical system typically involves learning to transfer data to the DAQ board in real-time by attaching a measurement sensor to a specific location on the hardware. However, this study intends to perform this task in the software space, where modeling and response calculation of the existing closed system multi-body system is assumed as the corresponding behavior. As implemented in previous studies, the following steps were followed: ① setting system degrees of freedom and constraint conditions, ② deriving differential-algebraic equations based on kinetic energy, potential energy, and constraint conditions, and ③ utilizing GNU Octave code that automatically calculates the response using the numerical integration method [24,31]. In order to show connectivity with previous studies, a 4-section mechanism was used for the target system of the closed system, and the design dimensions and initial conditions of each mechanism were applied identically (Fig. 3). Corresponding analysis modeling and response can derive response characteristics at that time by applying various states that can occur in actual hardware as it

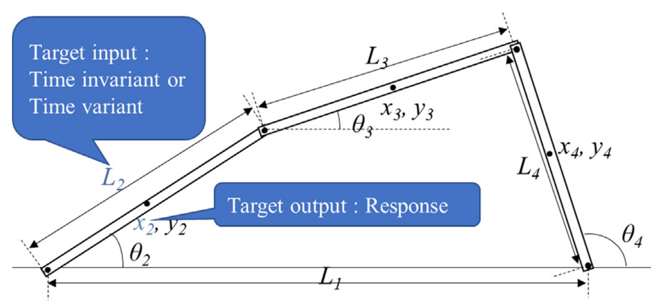

Fig. 3. 4-bar linkage.

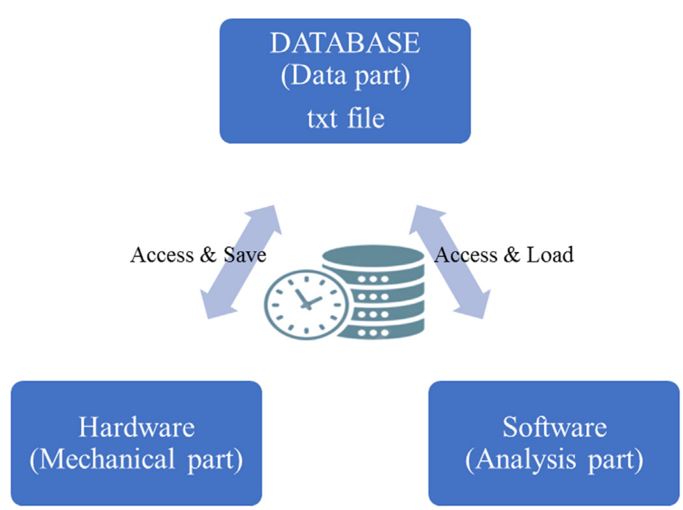

Fig. 4. Data interchange format.

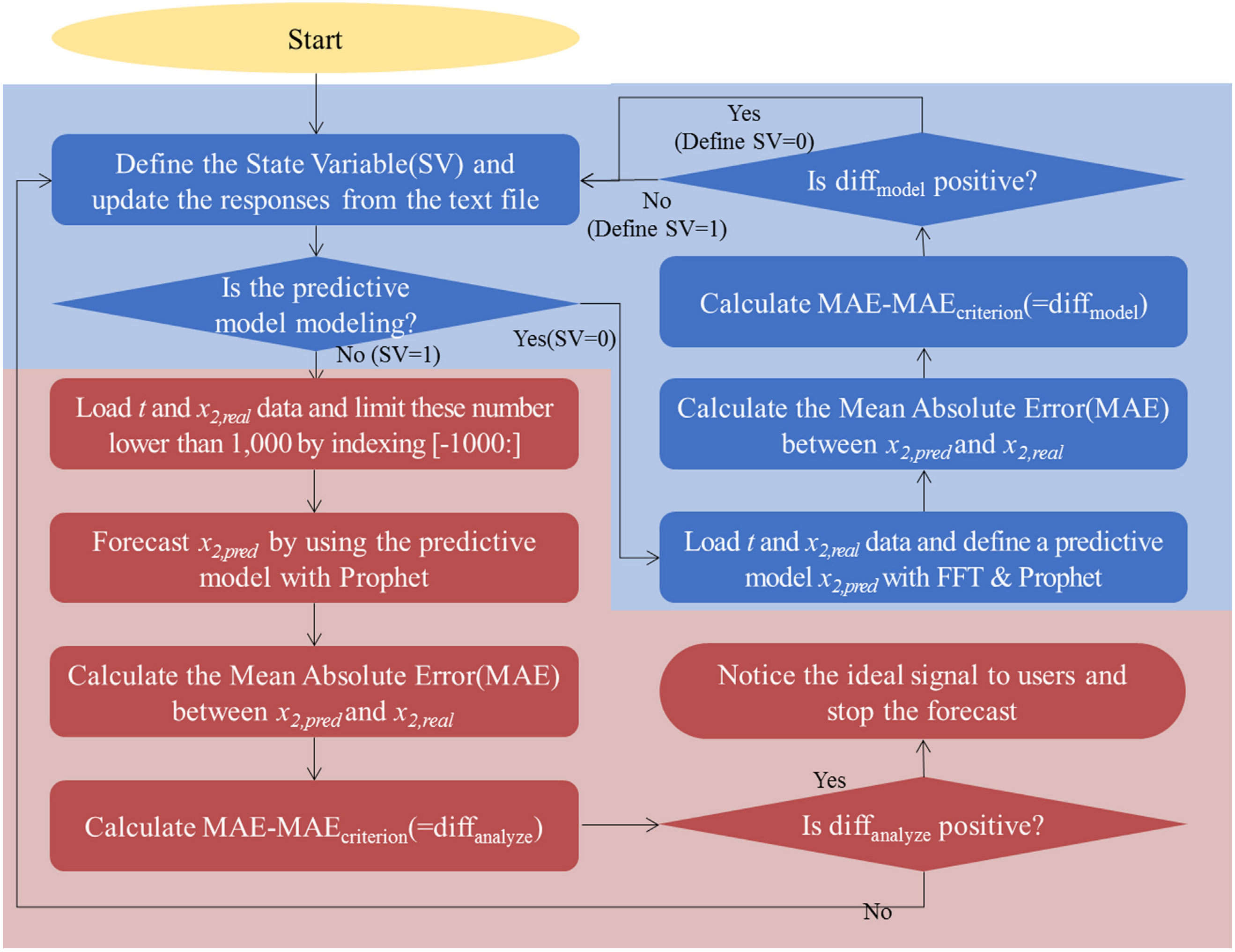

Fig. 5. Flow chart of the analysis part.

virtually simulates the response appearing in actual mechanical system hardware. Among various design dimensions and responses, how the response changes when the length L_2 of link 2 maintains a constant state or when the length of link 2 partially changes due to plastic deformation, etc., was designated as a target input to utilize, whether perceived or not. At this time, among the various responses to each link's translational and rotational motions, a structure was developed in which the position x_2 of link 2 was designated as a target output and input for the subsequent system.

3.3 Data part

To input/output the response result derived from an arbitrary device to other systems or devices, it is necessary to convert it to a compatible format. Signals generated from the GNU Octave mechanical part are in the form of double-type arrays, and quick data management is required within the program space. However, to be utilized by one of the subsystems, namely the Python data analysis unit, the response result needs to be converted into a compatible format. If I convert the data into formats such as xls, CSV, txt, etc., it can be accessed through the Python Dataframe. However, the critical point here is that the signal generated from the mechanical part is updated every hour, and it needs to be converted into a format that can access the data file in real-time and acquire the data. In other words, it is essential to quickly receive updated data from the analysis unit (Fig. 4) [32]. Considering these requirements, it was confirmed that the three subsystems could simultaneously perform file creation-access-load functions by processing time t and response x_2 in a text file. The GNU Octave data management unit uses the csvwrite command to input and save responses obtained every hour as cumulative text files.

3.4 Analysis part

The data analysis unit is a system that analyzes the response of the 4-bar mechanism stored as a computer file and informs the user of the inspection result for the current state. Since both the mechanical and data management departments produce data every hour, indicating the

Table 1. Design variable and simulation condition.

Case 1 for normal operating conditions of 4-bar linkage			
Design variable	Value	Initial & Simulation condition	Value
L_1	3 m	θ_2	30 deg
L_2	1 m	θ_3	20.71 deg
L_3	4 m	θ_4	49.98 deg
L_4	2.5 m	Other initial conditions	0 deg or deg/s
m_1	1 kg	T	0~100 sec
m_2	1 kg	Δt	0.01 sec
m_3	1 kg	Relative Tolerance	1×10^{-9}
g	$9.81\,\mathrm{m/s^2}$	Absolute Tolerance	1×10^{-6}
Case 2 for fault conditions of 4-bar linkage			
Design variable	Value	Initial & Simulation condition	Value
L_1	3 m	θ_2	30 deg
L_2	1→1.1 m(10 sec)	θ_3	20.71 deg
L_3	4 m	θ_4	49.98 deg
L_4	2.5 m	Other initial conditions	0 deg or deg/s
m_1	1 kg	T	0~100 sec
m_2	1 kg	Δt	0.01 sec
m_3	1 kg	Relative Tolerance	1×10^{-9}
G	$9.81\,\mathrm{m/s^2}$	Absolute Tolerance	1×10^{-6}

operation of the mechanical system, the data management department is responsible for: ① establishing a response prediction model and reviewing the reliability of the response data, ② reviewing the data updated every hour and providing the results to users, and ③ structuring a series of processes based on this scenario.

To implement the ①function of the data analysis unit, Python's Pandas Dataframe and Datareader were utilized for accessing txt files. Matrix processing was then conducted, followed by the establishment of a prediction model for response data, and finally, response analysis was performed using Prophet. During Prophet's fundamental trend analysis, it is optimized to mathematically express daily, monthly, and yearly trends. Therefore, after performing Fast Fourier Transform (FFT) independently, it is provided to Prophet as periodicity information to consider trend, periodicity, and specificity, thus aiding in establishing a predictive model. By utilizing the prediction model $x_{2,pred}$ (Eq. (13)) based on the real-time data x_2 from 0 s to t_{pred} when modeling is completed, the predicted data $x_{2,pred}$ for N times t_{pred}~t_{new}, newly updated after t_{pred}, and the real-time data $x_{2,real}$ are compared (Eq. (14)) to check for abnormalities. To implement the function, the sklearn library was used.

Additionally, a Mean Absolute Error (MAE) threshold was defined as the primary criterion for detecting abnormalities. If the MAE between the real-time $x_{2,real}$ and the predictive model $x_{2,pred}$ falls below the predefined threshold, the system marks the state variable as '1,' indicating a stable state, and stops further model updating. However, if the MAE exceeds the threshold, the system continues to monitor and model. The sensitivity of defect detection is dependent on the selection of the MAE threshold, which may vary depending on the system's operational characteristics. A higher threshold would make the system less sensitive to minor deviations, while a lower threshold increases the sensitivity, potentially detecting smaller faults. This flexibility allows the tool to adapt to different systems and conditions.

$$x_{2,pred} = g(t) + s(t) + h(t) + \varepsilon_t, \tag{13}$$

where $g(t)$ is a growth term, $s(t)$ is a seasonality term, $h(t)$ is holiday effects, and ε_t is a noise error term.

$$MAE = \frac{1}{N}\sum_{j=1}^{N}\left|x_{2,pred,i} - x_{2,real,i}\right| \tag{14}$$

To implement the ② function, it is necessary to create a loop so that the ① function can be repeatedly executed. Consequently, the code for ① was encapsulated into a function, and then the Schedule library was employed to schedule the repeated execution of ① at the set time intervals.

Regarding the final ③ function, each application point is depicted in Figure 5. The function was divided into a response prediction model modeling part in the blue area and a real-time response review part using the prediction model in the red area. Furthermore, an algorithm was developed to assess abnormalities based on the difference between the real-time $x_{2,real}$ and the established predictive model $x_{2,pred}$. If this difference exceeds the threshold at any point in time, the system identifies the presence of a defect; otherwise, it continues monitoring in a stable state.

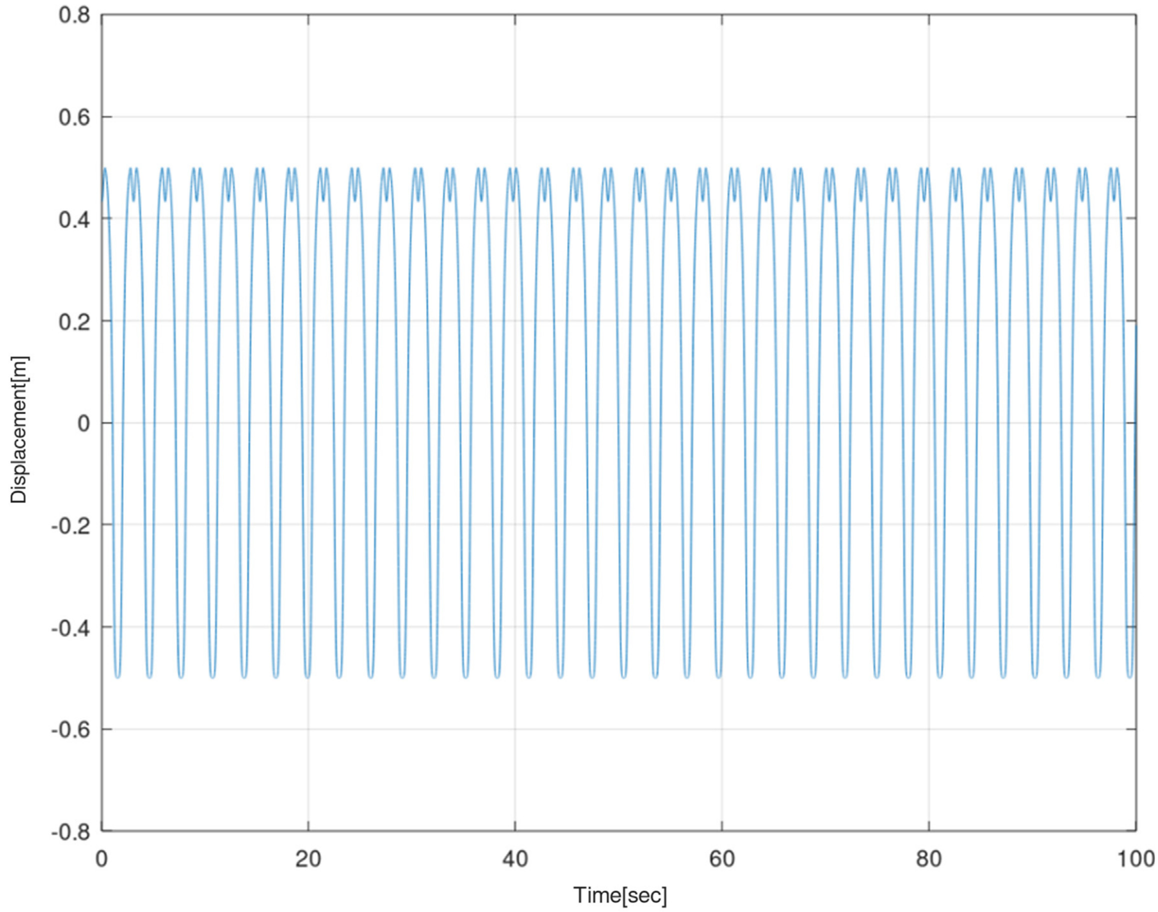

Fig. 6. Time history of x_2 response from $t = 0$ s to $t = 100$ s.

3.5 Integrated design system implementation scenario

To evaluate the performance and integration of the Integrated Design System, simulations were conducted using two primary scenarios. In the normal operating conditions scenario, Case 1, all design dimensions and operating conditions were maintained within their normal ranges. The Mechanical Part calculated the system response x_2 using the initial design dimensions and operational parameters defined for the 4-bar mechanism. This scenario was used to simulate stable operation and establish a baseline response for the mechanical system. The resulting time history of x_2 was then transferred to the Data Part, where it was converted into a compatible format (e.g., CSV) for further processing. The Analysis Part utilized this data to create a predictive model based on Fast Fourier Transform (FFT) and Prophet algorithms in real time, enabling the detection of trends, periodicities, and specific behaviors. Through iterative calculation, the Analysis Part finished modeling the predictive model to fit the experimental data from the Data Part.

To simulate fault conditions, Case2, the length of Link 2 (L_2) in the 4-bar mechanism was altered at an arbitrary time, representing a physical fault such as plastic deformation. The Mechanical Part calculated the system's response to reflect this change, generating updated time history data for x_2. This updated data was processed in the Data Part and analyzed in the Analysis Part by remodeling the predictive model to fit the new experimental data from Data Part before the fulat occurred. By comparing unexpected responses by the fault to the predictive model developed based on normal response partially prior to the fault, significant deviations between the predicted and real-time responses were identified, enabling the system to detect and classify the fault.

Table 1 summarizes the design dimensions and operating conditions applied in the analytical modeling process. Figure 6 illustrates the overall workflow for data exchange and fault detection in the integrated system. By simulating both normal and fault conditions, the Integrated Design System successfully demonstrated its potential to handle real-world applications, providing users with insights into system performance and enabling predictive maintenance.

4 Results and discussions

4.1 Time history of the mechanical part's response

Figure 6 depicts the time history of the x_2 coordinate for the center of link 2 from 0 to 100 s in the mechanical part of the four-bar linkage. The four-bar linkage is currently in the

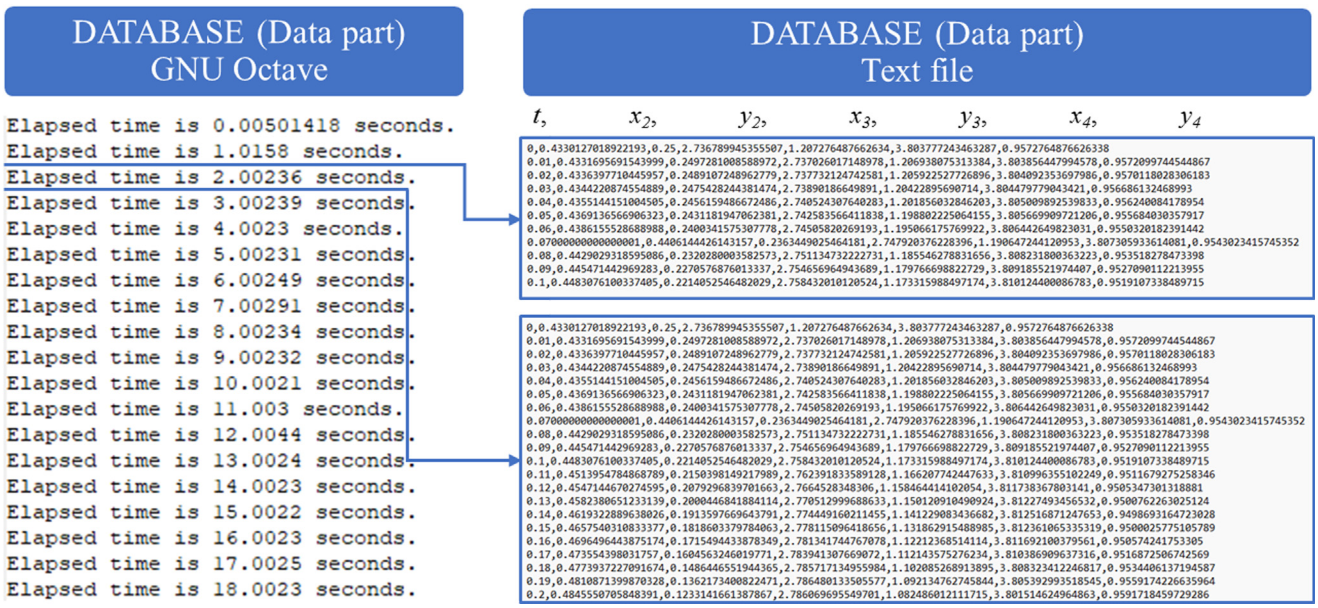

Fig. 7. Generation of the response data in real-time.

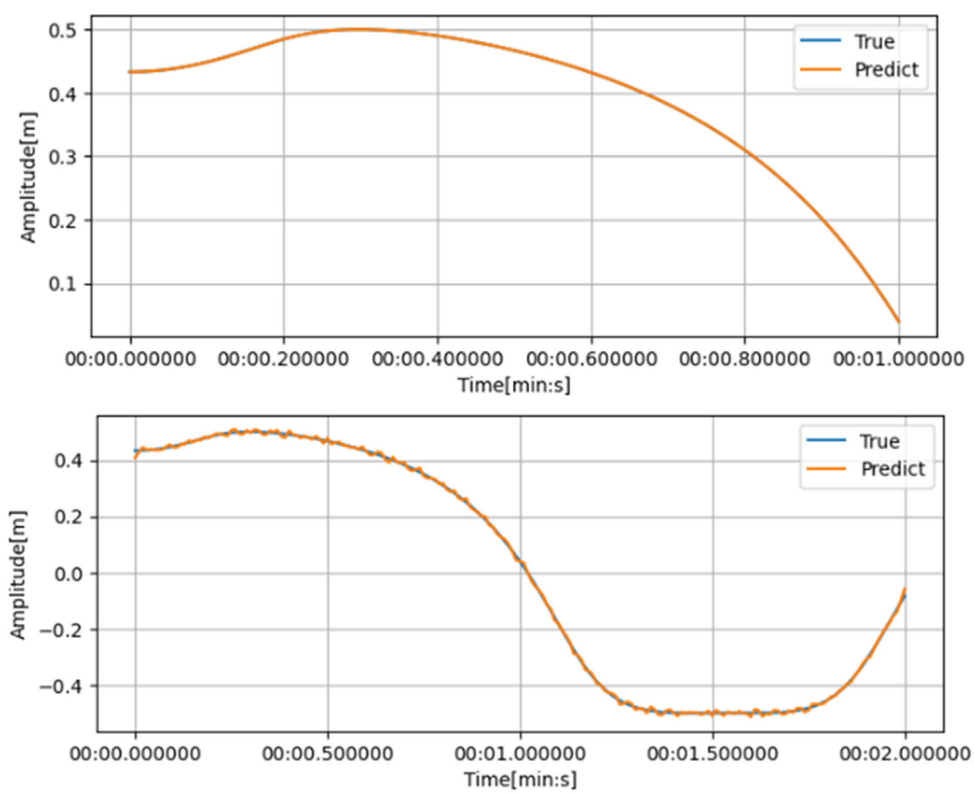

Fig. 8. Load the response data in real-time from 0 s to 1 s and from 0 s to 2 s, respectively.

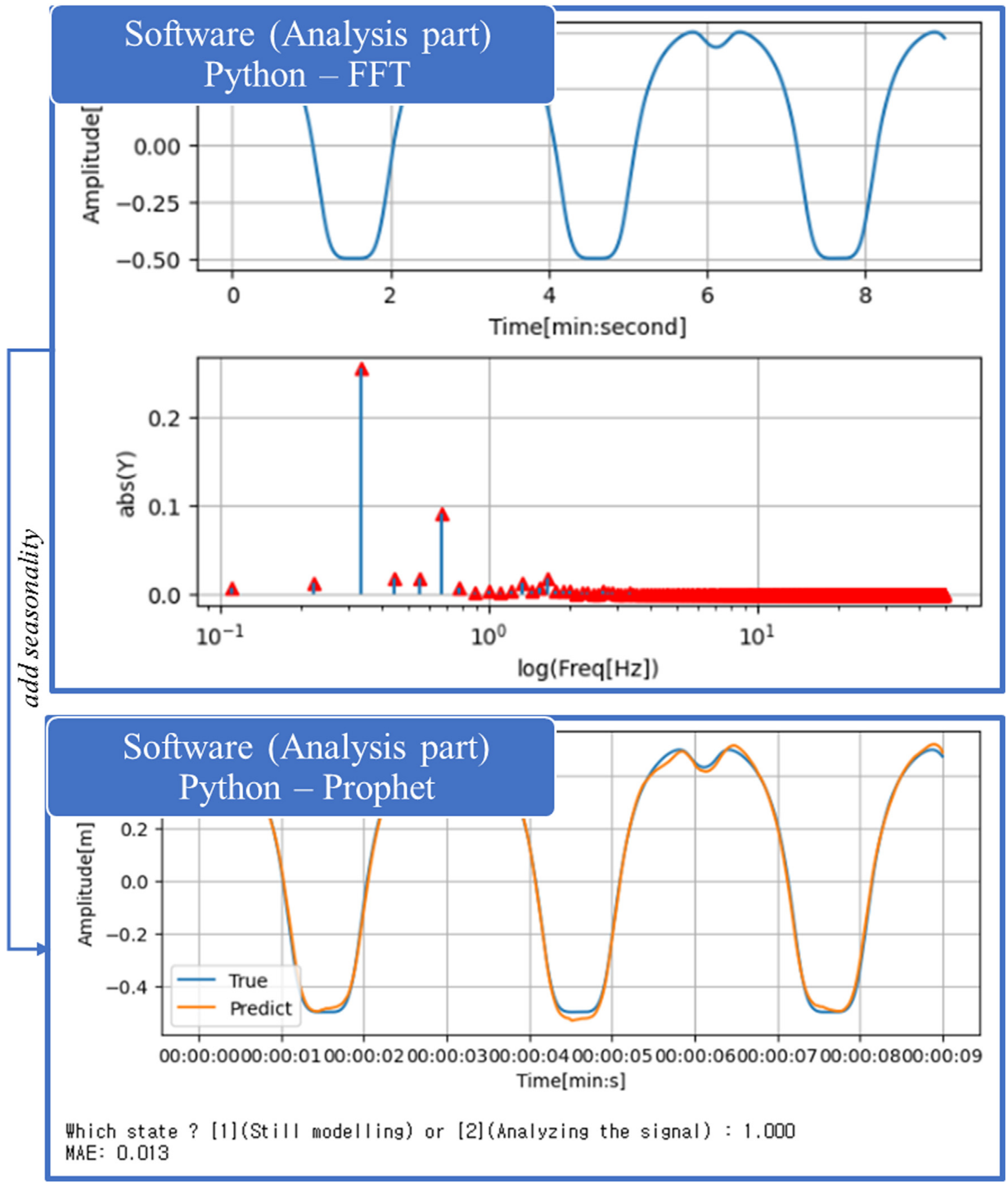

Fig. 9. Modeling the predictive model x_2 from $t=0$ s to 9 s using FFT and Prophet.

constrained system. It operates with a structure wherein the motion mechanism repeats according to the initial rotational displacement and speed of each link. Over time, the repeated x_2 response can be utilized to predict the state of the mechanical part through data processing, enabling the establishment and analysis of a response prediction model.

4.2 Data exchange and processing performance between subsystems

Figure 7 illustrates that the response time history derived from the mechanical part is updated as a text file every second in the data section. The response text file accumulates data in the sequence of time (t), x_2, y_2, x_3, y_3, x_4, y_4. This process continues until the completion of the mechanical part's response.

Figure 8 illustrates a modeling task that establishes a predictive model by accessing the data generated between $t=0$ and 1 s and $t=0$ and 2 s, as depicted in Figure 7, in real-time from the analysis unit. Through the interaction between time domain 1, updating real-time data generated in the mechanical part as a file, and time domain 2, automatically accessing the updated data in the analysis part, the lightweight text file capable of real-

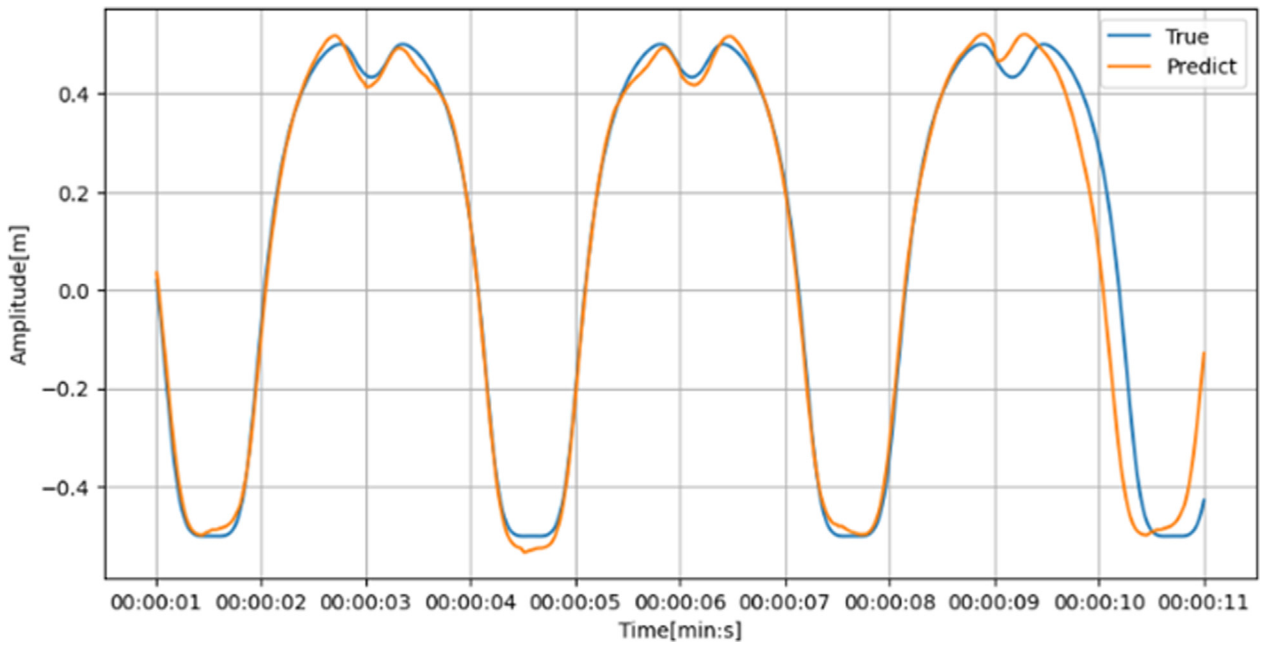

Fig. 10. Monitoring the mechanical system from $t = 0$ s to 11 s by comparing the actual responses to the predictive model.

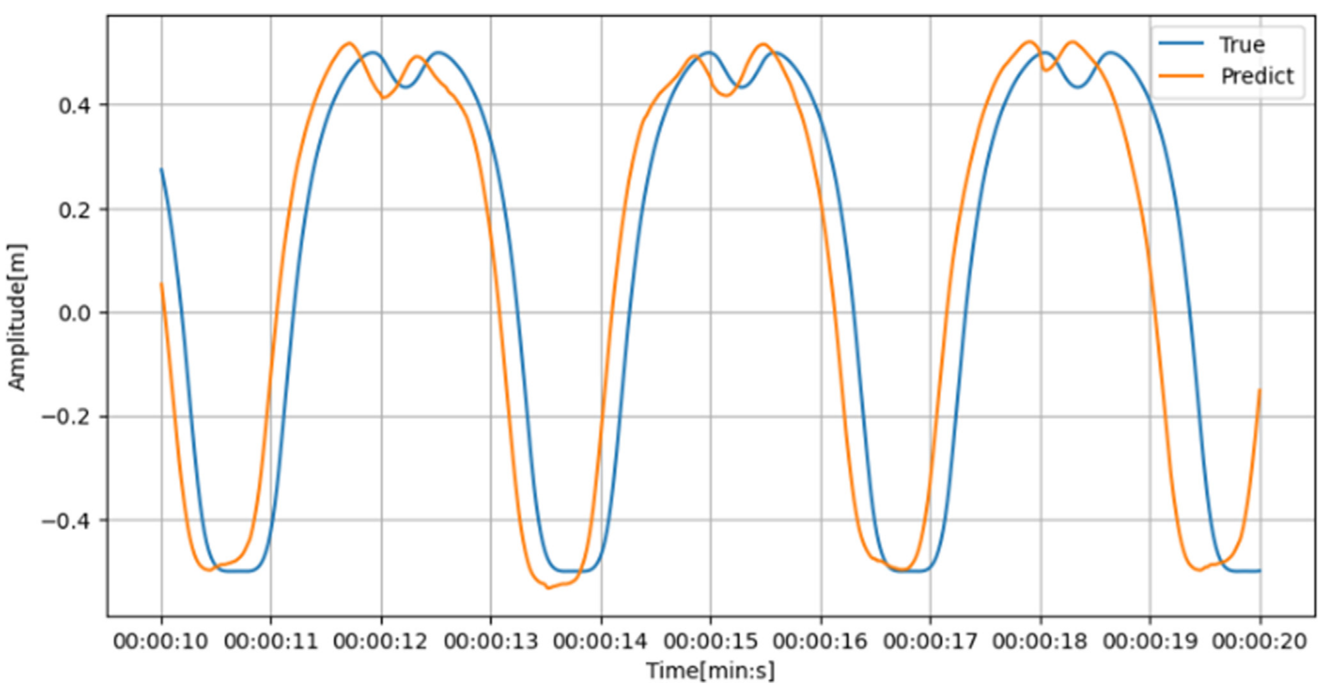

Fig. 11. Monitoring the mechanical system from $t = 10$ s to 20 s by comparing the actual responses to the predictive model.

time input-output facilitates targeted data exchange and processing functions, which were confirmed to be executed in real time.

4.3 Establishment of the response's prediction model in the analysis part

Figure 9 illustrates a modeling process in which the analysis unit utilizes real-time response data generated in the mechanical unit from $t = 0$ to 9 s to make predictions. By analyzing the frequency components of the signal using a Fast Fourier Transform and incorporating periodicity into the Prophet model, the accuracy of the prediction model can be enhanced, resulting in minimal differences (Mean Absolute Error – MAE) between the predicted model and the actual data.

Figure 10 presents a comparative review process where the predicted value $x_{2,pred}$ derived by the prediction model, based on the response data between $t = 0$ and 9 s, is compared with the actual response signal $x_{2,real}$ at the

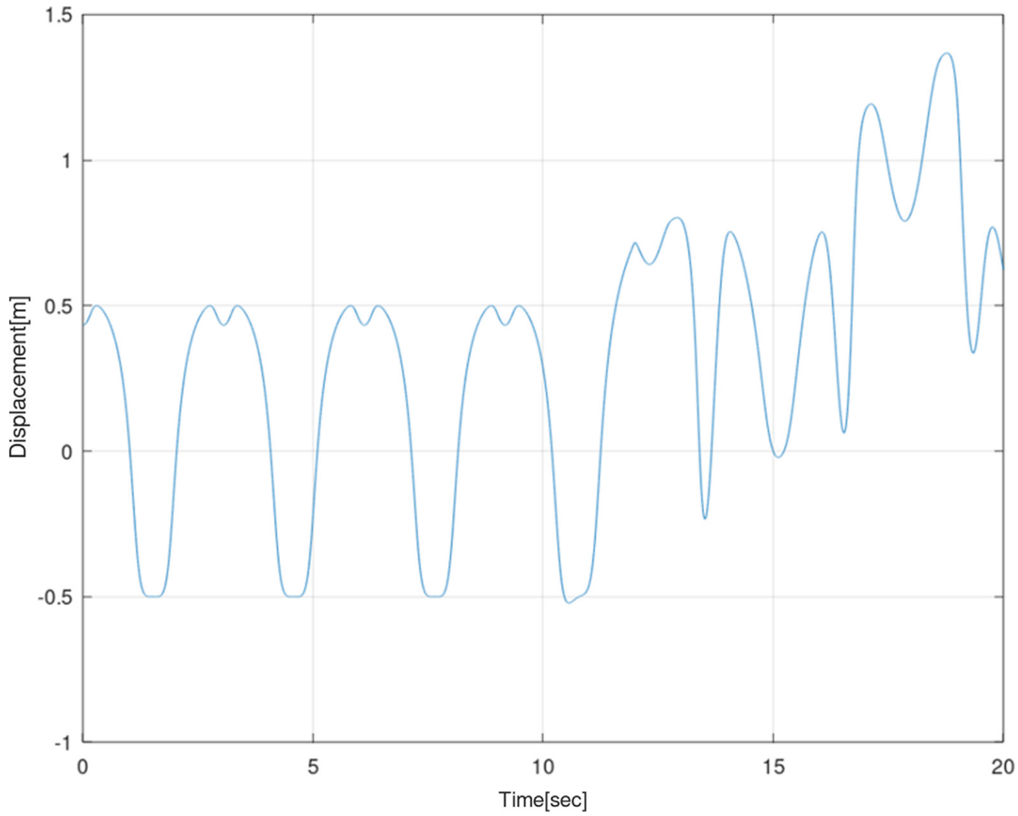

Fig. 12. Time history of x_2 response with the defect at $t = 10$ s.

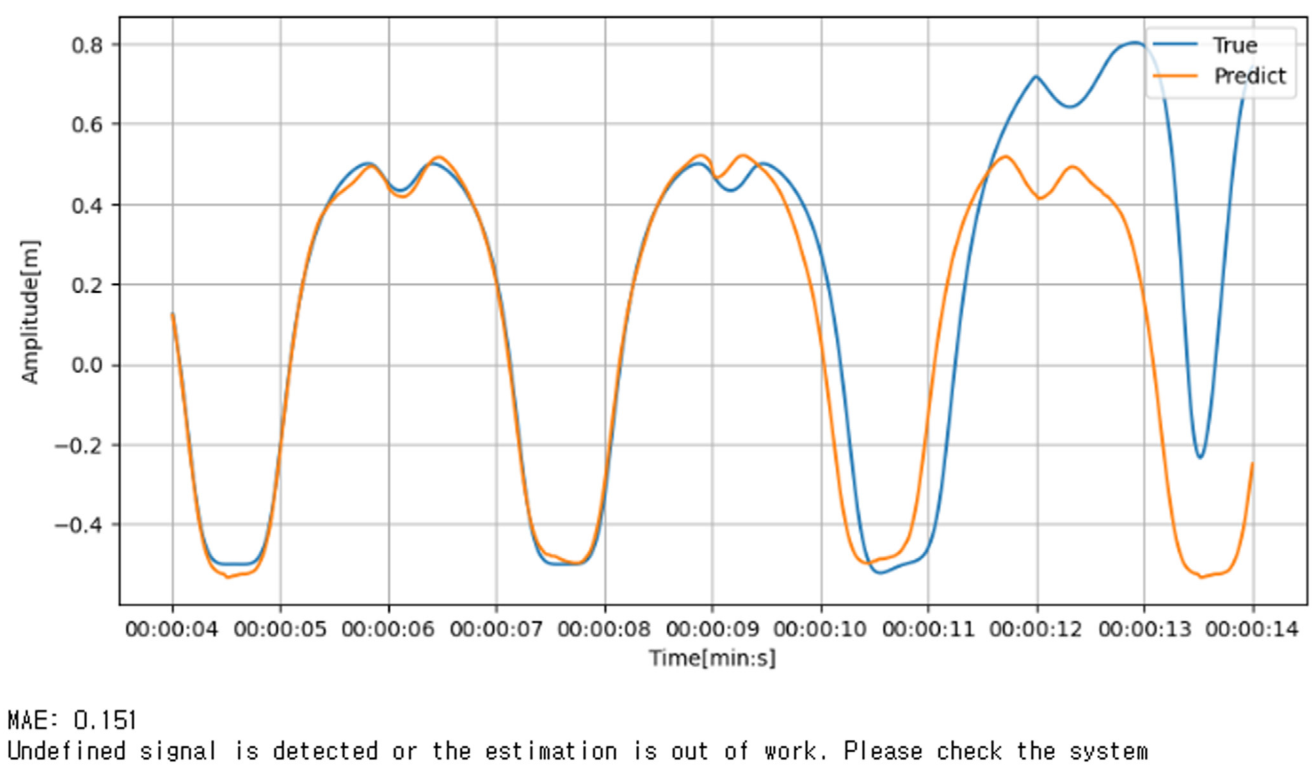

Fig. 13. Monitoring and alarming the defect from $t = 0$ s to 14 s.

updated time $t = 0$ and 11 s. The overall predictive model closely resembles the actual one in the graph. However, the reliability is somewhat compromised due to a predictive model with a specific value offset. In order to solve this problem, it can be presented as a linked learning case since it is necessary to increase the predictive degree by providing a sufficient learning opportunity for $t = 9$ s or more to the predictive model or to perform tuning work to increase the predictability of Prophet itself.

The number of data processed up to $t = 11$ s is 1001, and as the number increases significantly due to the accumulation of data or time, the performance deteriorates somewhat. In order to prevent this, the prediction model used to test the difference was applied by extracting only from the latest time entered in this prediction model to the time corresponding to 1000 times as a reference inverse. Figure 11 demonstrates that the difference between the actual value and the predicted model can be compared by using data from $t = 20$ s to $t = 10$ s, representing 1000 descending data points.

4.4 Fault detection performance based on abnormal state scenarios

Figure 12 depicts the time history of the rapid change in the x_2 response caused by the link 2 change fault at $t = 10$ s. Upon examining the monitoring performance to determine if it could be recognized after $t = 10$ s using the response data acquired up to 9 s, it was determined to be a definitive defect at $t = 14$ s.

The command to guide the user was derived, as shown in Figure 13. To expedite recognition, cognitive abilities can be enhanced through tasks such as improving the accuracy of the aforementioned predictive model, increasing the frequency of learning iterations, or enhancing sensitivity (tolerance for differences between actual data and current values).

5 Conclusion

In this study, an integrated engineering education model was developed that allows mechanical engineering students and industry personnel to experience a series of element technologies required for mechanical system modeling, control, and amortization in an integrated manner. Focusing on standard real-time response data, this study represents a significant convergence in education by incorporating machine learning technology necessary for developing intelligent products and production sites, including response simulation, data processing, data analysis, and notifications. This model exhibits features that can serve as exemplary cases. A challenge addressed in this study is the modeling process required for the intelligent implementation of mechanical systems characterized by repetitive signals. A model improvement method based on the latest machine learning predictive modeling technology, integrating deep learning, was implemented to address this issue, along with a method for faster data analysis. Future research endeavors aim to enhance learners' design capabilities throughout the entire cycle of machine design, from system modeling to control, by refining processing methods and detection scenarios. Furthermore, I intend to develop freely accessible code that facilitates dynamic analysis and simulation during the 3D modeling stage, thereby enhancing its usability in both industry settings and university dynamics analysis and system control data.

Funding

None.

Conflicts of interest

The author declared no potential conflicts of interest with respect to the research, authorship and publication of this article.

Data availability statement

All data generated or analyzed during this study are included in the present article.

Author contribution statement

The author, Yonghui Park, confirmed contribution to the paper as follows primarily; study conception and design, computational modeling, analysis and interpretation of results, draft manuscript preparation. In addition, the author reviewed the results and approved the final version of the manuscript.

References

[1] T.H. Yun, D.U. Seo, Securing manufacturing competitiveness for small and medium enterprises using open source software for engineering analysis, in: Proceedings of the Korea Technology Innovation Society Conference, Yeosu, Korea, 2015, p. 644–664
[2] C.H. Gu, Real-time software development technology trends, Curr. Ind. Technol. Trends Aerosp. **2**, 86 (2004)
[3] J.S. Kim, J.Y. Yu, H.Y. Kim, M.I. Kim, S.M. Lee, H. Han, S. U. Park, E.J. Kim, H.U. Park, H. Park, Technology Development and Application Supporting for Supercomputing Modeling & Simulation, Korea Institute of Science and Technology Information, Daejeon, 2016
[4] Y. Wei, The development and application of CFD technology in mechanical engineering, IOP Conf. Ser. Mater. Sci. Eng. **274**, 1 (2017)
[5] G.S. Morokina, U. Umbetov, Y.K. Mailybaev, Automation design systems for mechanical engineering and device node design, J. Phys. Conf. Ser. **1515**, 032061 (2020)
[6] C. Llopis-Albert, F. Rubio, S. Zeng, Multiobjective optimization framework for designing a vehicle suspension system. A comparison of optimization algorithms, Adv. Eng. Softw. **176**, 103375 (2023)

[7] C.B. Yu, B.K. Oh, Development of dynamic analysis automation system for constant velocity joint, in: Proceedings of the Korean Society of Automotive Engineers conference, Busan, Korea, 2022, p. 369
[8] H.J. Choi, J.R. Kim, H.G. Lee, Kinematic analysis of automatic tool changer change arm, in: Proceedings of the Korean Society of Manufacturing Process Engineers Conference, Yeosu, Korea, 2022, p. 289
[9] H.J. Choi, S.W. Kwon, Y.M. Jung, Dynamic analysis of arm part of automatic tool changer in vertical machining center, in: Proceedings of the Korean Society of Manufacturing Technology Engineers Conference, Gangneung, Korea, 2021, p. 129
[10] D.U. Jung, J.H. Cha, C.Y. Song, C.H. Lee, Development of an automation library in multi-body dynamics program for dynamic structural analysis of block lifting process, J. Soc. Nav. Archit. Korea, **53**, 135 (2016)
[11] E. Tandis, P. Cardiff, Finite volume-based supervised machine learning models for linear elastostatics, Adv. Eng. Softw. **176**, 103390 (2023)
[12] C.A.K.A. Kounta, L. Arnaud, B. Kamsu-Foguem, F. Tangara, Deep learning for the detection of machining vibration chatter, Adv. Eng. Softw. **180**, 103445 (2023)
[13] J. Kennedy, P. Hyland, A comparison of manufacturing technology adoption in SMEs and large companies, in: Proceedings of 16th Annual Conference of Small Enterprise Association of Australia and New Zealand, Ballarat, Australia, 2003, p. 1–10
[14] W.S. Yoo, Review of multibody dynamics, in: Proceeding of the Korean Society of Mechanical Engineers, Busan, Korea, 2009, p. 61–62
[15] W.S. Yoo, T.Y. Kim, S. Jung, History and future of multibody dynamics, Trans. Korean Soc. Mech. Eng. A, **43**, 483 (2019)
[16] N. Munro, Symbolic Methods in Control System Analysis and Design, IET, Stevenage, 1999
[17] N. Van Khang, Kronecker product and a new matrix form of Lagrangian equations with multipliers for constrained multibody systems, Mech. Res. Commun. **38**, 294 (2011)
[18] D. Negrut, D. Andrew, ADAMS/Solver Primer, MSC Software, Ann Arbor, 2004
[19] I.A. Bonev, Geometric Analysis of Parallel Mechanisms, Université Laval, Qubec, 2002
[20] L. Schwab, M. Wisse, Lecture Notes Multibody Dynamics B, Delft University, Delft, 1998
[21] S.H. Sadati, S.E. Naghibi, M. Naraghi, An automatic algorithm to derive linear vector form of Lagrangian equation of motion with collision and constraint, Procedia Comput. Sci. **76**, 217 (2015)
[22] S.H. Sadati, S.E. Naghibi, A. Shiva, S. Zschaler, H. Hauser, I. Walker, K. Althoefer, T. Nanayakkara, AutoTMTDyn: A Matlab software package to drive TMT Lagrange dynamics of series rigid-and continuum-link mechanisms, in: IROS 2018 Workshop on Soft Robotic Modeling and Control: Bringing Together Articulated Soft Robots and Soft-Bodied Robots, Madrid, Spain, 2018
[23] Y.H. Park, An automatic program of generation of equation of motion and dynamic analysis for multi-body mechanical system using GNU octave, J. Appl. Comput. Mech. **7**, 1687 (2021)
[24] Y.H. Park, Development of an educational code of deriving equations of motion and analyzing dynamic characteristics of multibody closed chain systems using GNU octave for a beginner, J. Appl. Comput. Mech. **8**, 232 (2022)
[25] H.H. Yoo, Kane's methodology for deriving the equations of motion, Trans. Korean Soc. Noise Vib. Eng. **14**, 40 (2004)
[26] K.D. Yang, D.S. Bae, S.M. Yang, C.K. Choi, An index 2 differential-algebraic equation formulation for multibody system dynamics, Trans. Korean Soc. Mech. Eng. **19**, 2769 (1995)
[27] E. Fehlberg, Classical fifth-and seventh-order Runge-Kutta formulas with stepsize control, Computing, **4**, **93** (1969)
[28] G.Y. Luo, D. Osypiw, M. Irle, Vibration modelling with fast Gaussian wavelet algorithm, Adv. Eng. Softw. **33**, 191 (2002)
[29] X. Chu, W. Cui, S. Xu, L. Zhao, H. Guan, Y. Ge, Multiscale time series decomposition for structural dynamic properties: long-term trend and ambient interference, Struct. Control Health Monit. 2023, **6485040** (2023)
[30] S.J. Taylor, B. Letham, Forecasting at scale, AM STAT, **72**, 37 (2018)
[31] M. Arda, Dynamic analysis of a four-bar linkage mechanism, Mach. Technol. Mater. **14**, 186 (2020)
[32] Y.M. Altman, Accelerating MATLAB Performance: 1001 Tips to Speed up MATLAB Programs, CRC Press, Boca Raton, 2014

Cite this article as: Y. Park, Development of an integrated engineering education model using multi-body closed system modeling, computational analysis, and signal processing analysis, Mechanics & Industry **26**, 16 (2025) https://doi.org/10.1051/meca/2024035