IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 October 2022, accepted 14 November 2022, date of publication 18 November 2022,
date of current version 2 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3223182

RESEARCH ARTICLE

A Framework for Data Driven Dynamic Modeling of Serial Manipulators

MOHAMED OMAR[1,2], **RUIFENG LI**[1], **AND AHMED ASKER**[2,3]
[1]State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China
[2]Production Engineering and Mechanical Design Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[3]School of Engineering and Applied Sciences, Nile University, Giza 12573, Egypt

Corresponding author: Ruifeng Li (lrf100@hit.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62176072.

ABSTRACT Robot dynamic modeling and parameter identification are essential for many analyses. High-fidelity multi-body dynamics simulators can model the robot's dynamic behavior, but they can't identify the robot's non-linear dynamic model needed for controller design. This study proposes a three-step machine-learning framework for extracting the dynamic equations of serial manipulators from data. This framework consists of three steps. Initially, a library of candidate functions is constructed, together with a data set based on the robot's unforced response. Secondly, the best models that can represent dynamic systems for each candidate function utilizing the training data set are then obtained using the SINDy-PI algorithm and Akaike Information Criterion (AIC). Through the MSE and test data, those best models will be reduced to get the functions that best describe the dynamic system. Finally, the dynamic equations that characterize the system are derived using the SINDy algorithm, including the applied force or torque. The proposed framework was tested on three case studies —double pendulum, two-link of KUKA robot, and two-link of Stanford robot. The framework correctly determines the structure of the dynamic system and simultaneously accurately identifies its parameters. The framework was able to deal with an ill-conditioned system of equations that arises for complex robot configuration.

INDEX TERMS Dynamic equations, SINDy, SINDy-PI, serial manipulators, data-driven equations, machine learning.

I. INTRODUCTION

Dynamic modeling is one of the fundamental problems in robotics. It provides joint reaction forces and moments, which are needed for links, bearings, and actuation design/selection [1], [2]. Dynamic models are a prerequisite for designing model-based controllers, which are the most common type of controllers in the industry [3]. Traditionally, differential equations of motion are derived by conventional methods such as Newton-Euler and Euler-Lagrange methods [1]. The derived model by those methods is as good as the assumption used to simplify the model. Applying those methods to complex robot configurations can be quite challenging [4]. Furthermore, parameter identification is needed to identify the model parameter based on the obtained model. Multi-body dynamics simulators such as MSC ADAMS®, Simscape Multibody and Gazebo can perform accurate, realistic simulations of robotic systems. These simulators can use the 3D model or the Unified Robot Description Format (URDF) file to define the geometry as well as the kinematic and dynamic parameters of the robot. Moreover, it can provide control design tools, but these tools depend on the linearized dynamic model. That makes these control design tools unsuitable for the design of high-performance controllers [5]. There is an increasing interest in learning the governing physical laws from data, which is motivated by the recent advances in modern data-driven techniques—machine learning [6] and big data [7] —and the availability of an abundance of data [8]. Several approaches for extracting differential equations (dynamic model) from time series data have been presented,

The associate editor coordinating the review of this manuscript and approving it for publication was Yangming Li.

like the Gaussian process [9], [10], compressive sensing [11], [12], [13], neural network [14], [15], and sparse regression [16], [17]. These approaches were largely employed in the nonlinear identification of fluid systems [18]. Neural networks are used to find relations between input and output. In [14], they employed neural networks that have been trained to solve supervised learning tasks while respecting any physics laws described by nonlinear partial differential equations [15]. Regrettably, neural networks are a black box model that does not reveal much about how the input and the output are related. Compared to black boxes, interpretable models are preferable [8], [19].

Symbolic regression was used in [20] to develop an interpretable but sufficiently accurate model. That demonstrated the feasibility of inferring natural rules from experimental data. It was also proposed that the learning rate may be increased by bootstrapping the learning process using the laws from a simpler system. Sparse identification of nonlinear dynamics [21] (SINDy) has become a useful tool for identifying underlying dynamics. This approach simulates dynamics by combining nonlinear basis functions into a linear combination. It then employs a sparse regression method to identify the smallest number of terms necessary for the dynamic governing equations to accurately represent the data. SINDy has been modified to deal with different problem formulations, e.g., SINDy-PDE (SINDy for Partial Differential Equations) [17], SINDy-c (Sparse Identification of Non-Linear Dynamics with control) [16], implicit-SINDy [22], abrupt-SINDy [23], PI-SINDy (Parallel Implicit Sparse Identification of Nonlinear Dynamics) [24], modified-SINDy [25] and Lagrangian-SINDy [8].

Serial manipulators yield a highly non-linear dynamic model. Moreover, the effect of the external forces/payloads applied to the end-effector needs to be considered. It is challenging to compute the relationship between force/torque and kinematic parameters from the Lagrangian function obtained by Lagrangian-SINdy method [4]. Most of the aforementioned techniques are applied to simple geometries. Also, the obtained model is not in its generalized form. For example, the SINDy-PI algorithm did not consider the actuation force and needed an explicit expression for joint acceleration. Thus, the effect of the external forces/payloads needs to be considered as part of the problem, which increases the number of terms to be identified. In [4], a method inspired by SINDy is proposed to determine the torque equation based on Lasso regression. However, no comparison between the obtained model terms and the closed-form solution was reported. Also, a robot with simple geometry and only a revolute joint was studied.

This work is motivated by the difficulties faced when applying SINDy to industrial serial manipulators, which have a more complex configuration than the double pendulum model obtained in [24]. Moreover, the method proposed in [4] did not adequately perform when applied to robots with geometry, which are found in the industry. A robot with complex geometry has more terms in the governing dynamic equation, which consequently leads to an ill-conditioned matrix [24]. That greatly degrades the accuracy of the solution.

This research tackles three main problems. Firstly, the need to choose a suitable trajectory that excites the dynamic of the robot. Secondly, the need to deal with an ill-conditioned system of equations without degrading solution accuracy. Thirdly, the need to formulate the dynamic model in the generalized form to incorporate the effect of the external forces and the joint forces/torque. To address these problems, we propose a general framework inspired by SINDy algorithm. The library of candidate functions is reduced to the most affecting functions using SINDy-PI based on the unforced motion of the robot. This eliminates the need for choosing a special trajectory that excites the dynamic of the robot. Then, using the reduced library, the dynamic model of the manipulator will be calculated through the SINDy algorithm. The problem is formulated such that the obtained model is in the generalized form. Also, considering both the unforced and forced responses of the system enables the framework to deal with an ill-conditioned system of equations without sacrificing accuracy. The framework is applied to different configurations of serial manipulators with 2 degrees of freedom DoFs to make it comparable with previous work performed in [24].

The remaining of this paper is organized as follows. The methodology is described in Section II. The results and discussions are demonstrated in Section III. Concluding remarks are presented in Section IV.

II. METHODOLOGY

A. EULER-LAGRANGE METHOD

The Euler-Lagrange method will be briefly reviewed here for the sake of completeness. It will be used to derive the closed-form solution for the robots considered in this study. That will be used as a base for comparison with the dynamic equation obtained by the proposed approach.

In general, the application of the Euler-Lagrange equations to a robot with n joints yields a system of n coupled, second order nonlinear ordinary differential equations. The first step is to formulate the Lagrangian function, L, which is the difference between the kinetic K and potential P energy, Equation 1.

$$L = K - P \tag{1}$$

The dynamic governing equation of a serial robot can be obtained as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L}{\partial \boldsymbol{q}} = \boldsymbol{\tau} \tag{2}$$

The obtained dynamic equation (2) is a linear combination of linear and non-linear functions of $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, where $\boldsymbol{q} \in \Re^n$, $\dot{\boldsymbol{q}} \in \Re^n$ and $\ddot{\boldsymbol{q}} \in \Re^n$ are the generalized coordinate, generalized velocity and generalized acceleration vectors, respectively; while $\boldsymbol{\tau} \in \Re^{\mathrm{n}}$ is the generalized torque/force vector, which can be expressed in terms of joint torque vector,

$\boldsymbol{\tau}_q \in \Re^n$, and the vector of the force and moment exerted at the end-effector, $\boldsymbol{\tau}_e \in \Re^6$, as follows [1]:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{q}} + J^T \boldsymbol{\tau}_e \tag{3}$$

where $J \in \Re^{6\times n}$ is the Jacobian matrix and $\boldsymbol{\tau_e} = \left[\boldsymbol{F^T}\ \boldsymbol{M^T}\right]^{\boldsymbol{T}} = \left[F_x\ F_y\ F_z\ M_x\ M_y\ M_z\right]^T$. In which $\boldsymbol{F}$ and $\boldsymbol{M}$ are the applied force and moment, respectively specified in the base coordinate frame.

B. PROPOSED FRAMEWORK

The proposed framework is implemented in a three-stage algorithm, as shown in Fig.1. Those stages are library generation, library reduction, and final torque equations determination.

1) STEP (I) – LIBRARY GENERATION

The library is a combination of candidate functions that may appear in the differential equations of the robot. In the formulation of the SINDy-Lagrangian method [8], it was shown that the library for serial robots is made up of constants, trigonometric functions, and polynomials as elementary functions. Also, the identification of more complicated systems can be facilitated by the previous knowledge obtained from simpler systems. In this research, the candidate functions are chosen as a combination of state derivatives and trigonometric functions.

To generate the training data, the robot model will be represented by a URDF file, or by commercial CAD software like SOLID WORKS or CATIA. Then, the model will be imported into MATLAB. The parameters of the stimulation were adopted from [24]. The model is simulated for 10 seconds with a time step of 0.001 seconds to get the training data ($\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$).

2) STEP (II) – LIBRARY REDUCTION

The functions library Q as well as the training data will be combined to obtain $\boldsymbol{\theta}(\mathrm{Q}) \in \Re^{m\times p}$, where m is the number of observations in simulation data and p is the number of candidate functions in the library, Q. SINDy-PI [22] uses the matrix $\boldsymbol{\theta}(\mathrm{Q})$ to reduce the library Q. The free response is used in this step. Since the proposed framework is formulated using generalized coordinates, the forced response can be used in step III without modifying the library.

a: SINDy-PI

The mathematical model of SINDy-PI is:

$$\theta_j(Q) = \boldsymbol{\theta}(Q \mid \theta_j(Q))\xi_j \tag{4}$$

where $\boldsymbol{\theta}(Q \mid \theta_j(Q)) \in \Re^{m\times p-1}$ is the library $\boldsymbol{\theta}(Q)$ with the $\theta_j \in \Re^{m\times 1}$ column removed. The sequential threshold approach will be used to obtain the sparse coefficient vector ξ_j, [21]. The *cond* function in MATLAB is used to assess the condition number. When the matrix is well-conditioned, the condition number is close to 1 and the Least Squares solution will be used as in [21].

$$\underset{\xi_j}{min}\ ||\ \theta_j(Q) - \boldsymbol{\theta}(Q \mid \theta_j(Q))\xi_j\ ||_2 + \lambda||\ \xi_j\ ||_0 \tag{5}$$

where λ is the sparsity promoting parameter. When the condition number is much larger than 1, the matrix becomes ill-conditioned. The least norm solution is used in the proposed approach, since it provide a more robust solution in this case [26]. It is implemented in MATLAB using the pseudoinverse function, *pinv*.

$$\underset{\xi_j}{min}\ ||\ \xi_j\ ||_2 + \lambda||\ \xi_j\ ||_0 \tag{6}$$

Equation 5 or 6 will be solved for each candidate function in the library. It is necessary to test for each θ_j because the structure of the dynamic equation is not known at this stage.

The precision of the solution for coefficients ξ is not assured regardless of the method employed to solve an ill-conditioned matrix, i.e., Equation 4. The proposed approach tackles this problem by using this step only to reduce the library, and the solution for coefficients is implemented in step III. In contrast to the least square solution, the least norm solution demonstrated a better ability to identify the most promising candidate functions.

b: MODEL SELECTION

The solution of Equation 5 or Equation 6 (Ill-conditioned matrix) yields several models for each candidate function θ_j. This results from solving the system for different thresholds, λ, settings. When the candidate function θ_j is not present in the actual dynamics, the derived coefficient vector ξ_j will not be sparse, and there will be a significant prediction error. Choosing the best candidate function results in a sparse coefficient vector and a small prediction error. So, for each function, the best model over different threshold values must be selected, and to do so, the Akaike information criterion (AIC) [27], [28], [29], [30] and Pareto model [27] will be employed. AIC score can be calculated using the following equation.

$$AIC = 2k - 2\ln(SSE) \tag{7}$$

where k is the number of free parameters to be estimated and SSE is sum of squared error of the model for each λ and is calculated as follows,

$$SSE = \left(\sum_{i=1}^{m} \theta_j(Q) - \boldsymbol{\theta}(Q \mid \theta_j(Q))\xi_j\right)^2 \tag{8}$$

For each candidate function θ_j of the library, the AIC score for each λ and number of active terms of the model will be used to build Pareto front [27]. For each candidate function, the model with the lowest AIC score will be chosen.

The next task is to decide which of the models we found for each candidate function best captures the dynamics of the robot. The test data is obtained by simulating the system for

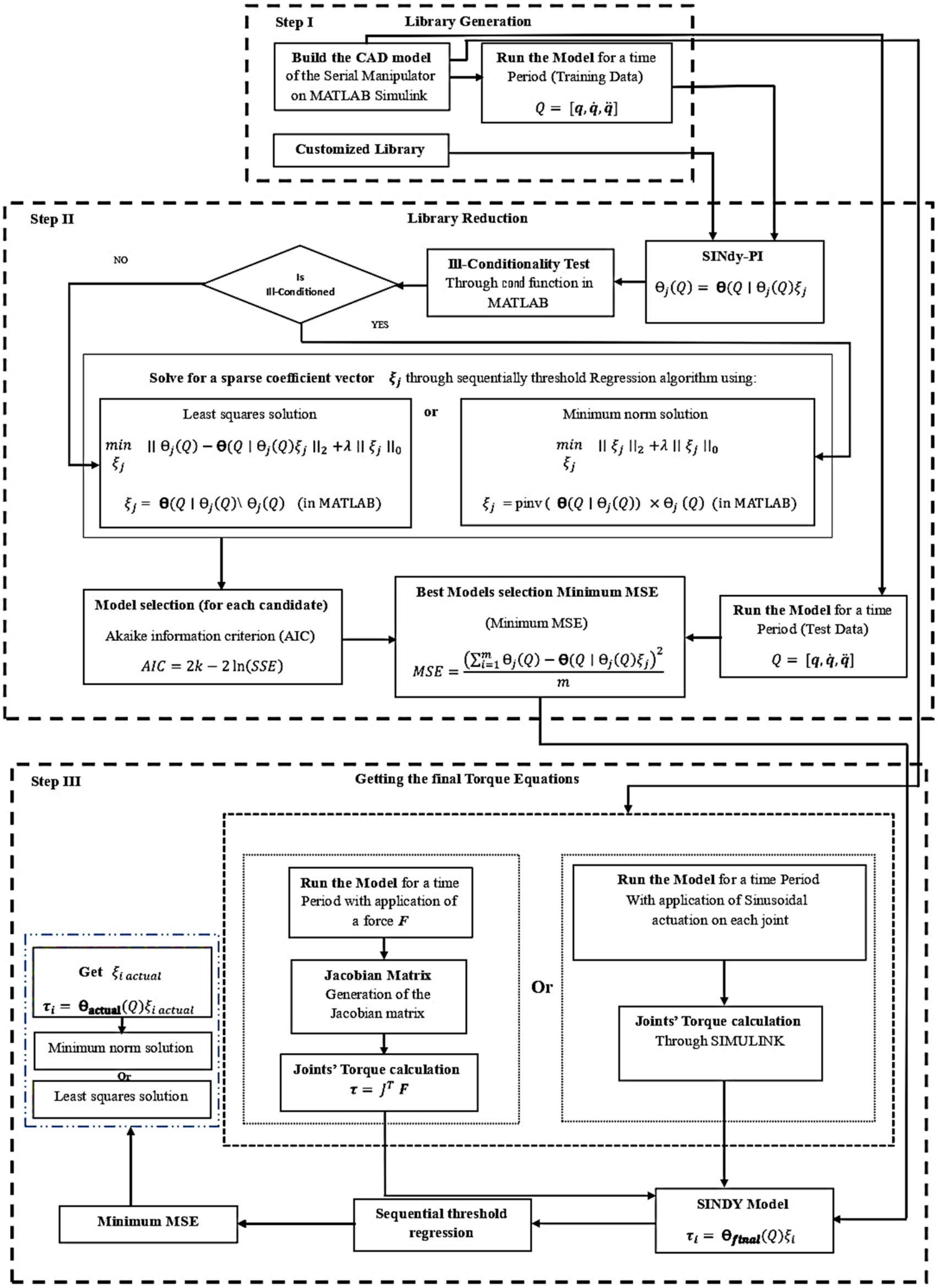
Step I
Library Generation
Build the CAD model of the Serial Manipulator on MATLAB Simulink
Run the Model for a time Period (Training Data)
$Q = [q, \dot{q}, \ddot{q}]$
Customized Library
Step II
Library Reduction
NO
Is Ill-Conditioned
YES
Ill-Conditionality Test Through cond function in MATLAB
SINdy-PI
$\Theta_j(Q) = \mathbf{\Theta}(Q \mid \Theta_j(Q)\xi_j$
Solve for a sparse coefficient vector ξ_j through sequentially threshold Regression algorithm using:
Least squares solution
$\min_{\xi_j} \| \Theta_j(Q) - \mathbf{\Theta}(Q \mid \Theta_j(Q)\xi_j \|_2 + \lambda \| \xi_j \|_0$
$\xi_j = \mathbf{\Theta}(Q \mid \Theta_j(Q) \backslash \Theta_j(Q)$ (in MATLAB)
or
Minimum norm solution
$\min_{\xi_j} \| \xi_j \|_2 + \lambda \| \xi_j \|_0$
ξ_j = pinv($\mathbf{\Theta}(Q \mid \Theta_j(Q))$ × $\Theta_j(Q)$ (in MATLAB)
Model selection (for each candidate)
Akaike information criterion (AIC)
$AIC = 2k - 2\ln(SSE)$
Best Models selection Minimum MSE
(Minimum MSE)
$MSE = \frac{(\sum_{i=1}^{m} \Theta_j(Q) - \mathbf{\Theta}(Q \mid \Theta_j(Q)\xi_j)^2}{m}$
Run the Model for a time Period (Test Data)
$Q = [q, \dot{q}, \ddot{q}]$
Step III
Getting the final Torque Equations
Run the Model for a time Period with application of a force F
Jacobian Matrix Generation of the Jacobian matrix
Joints' Torque calculation
$\tau = J^T F$
Or
Run the Model for a time Period With application of Sinusoidal actuation on each joint
Joints' Torque calculation Through SIMULINK
Get $\xi_{i\,actual}$
$\tau_i = \mathbf{\Theta}_{\mathbf{actual}}(Q)\xi_{i\,actual}$
Minimum norm solution
Or
Least squares solution
Minimum MSE
Sequential threshold regression
SINDY Model
$\tau_i = \mathbf{\Theta}_{\boldsymbol{final}}(Q)\xi_i$

FIGURE 1. Proposed framework.

3 seconds with a time step of 0.001 seconds. This test data is then utilized to evaluate each model using MSE, Equation 9.

$$MSE = \frac{\left(\sum_{i=1}^{m} \theta_j(Q) - \boldsymbol{\theta}(Q \mid \theta_j(Q)\,\xi_j\right)^2}{m} \tag{9}$$

For nonredundant serial robots, the number of joints determines how many differential equations there are for serial manipulators. For instance, a 2 DOF robot will need a system of two differential equations to describe its dynamics. The robot's governing equations of motion will be selected from the models obtained for each candidate function. The selection process depends on two criteria: minimum MSE and at least two differential equations have the same terms. The MSE measure (Equation 9) used in the proposed framework differs from the error selection function used in [24].

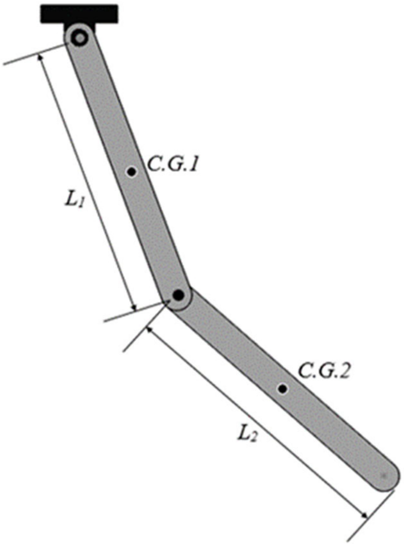

FIGURE 2. 2D Schematic illustration of the double pendulum.

3) STEP (III) – FINAL EQUATIONS

When the two criteria mentioned in Section 2.B.II were achieved, the differential equations that were found in the earlier steps take the following form:

$$\theta_{final}\,(Q) = \boldsymbol{\theta_{final}}(Q \mid \theta_{final}\,(Q))\xi_{final} \tag{10}$$

where, after decreasing the library, θ_{final} is a function of the remaining candidate functions. All candidate functions other than the function θ_{final} are contained $\boldsymbol{\theta_{final}}(Q \mid \theta_{final}\,(Q))$. Equation 10 cannot be used to plan and control the trajectory of serial manipulators as it is not in the generalized form. They must have the following form:

$$\tau_i = \boldsymbol{\theta_{final}}\,(Q)\,\xi_i \tag{11}$$

where τ_i is the i$^{\text{th}}$ element of the generalized torque vector $\boldsymbol{\tau}$.

In this step, there are many ways to determine the joint torque τ_i in Equation 11—one possible method is to simulate the system motion as a result of applying a generalized force to the end-effector, and another possible method is to apply a certain trajectory in the joint space and calculate the required actuation torque. Applying a generalized force to the end effector and a trajectory in the joint space is also possible, as shown in Equation 3.

a: APPLICATION OF GENERALIZED FORCE

The system is operated for 3 seconds with a time step of 0.001 seconds and force $\boldsymbol{\tau_e}$ applied to the end-effector and $\boldsymbol{\tau_q} = 0$ to capture a new dataset $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\tau})$. The Jacobian matrix J is also calculated as it is used to map between the generalized external force $\boldsymbol{\tau_e}$ and the generalized torque as follows:

$$\boldsymbol{\tau} = J^T\boldsymbol{\tau_e} \tag{12}$$

TABLE 1. Input parameters of case study I.

Parameter	Value
L_1	0.27 *m*
L_2	0.285 *m*
m_1	0.942 *kg*
m_2	0.942 *kg*
I_1 (Inertia of link 1) $kg.m^2$ (Related to frame at C.G)	$\begin{bmatrix} 0.0071906 & 0 & 0 \\ 0 & 0.00707285 & 0 \\ 0 & 0 & 0.00013345 \end{bmatrix}$
I_2 (Inertia of link 2) $kg.m^2$ (Related to frame at C.G)	$\begin{bmatrix} 0.0071906 & 0 & 0 \\ 0 & 0.00707285 & 0 \\ 0 & 0 & 0.00013345 \end{bmatrix}$
g (Gravity)	$[0\ {-9.80665}\ 0]^T\ m/s^2$

FIGURE 3. KUKA Robot: KR 4 AGILUS [31].

b: APPLICATION OF TRAJECTORY IN THE JOINT SPACE

The system is operated for 3 seconds with a time step of 0.001 seconds, with the application of a sinusoidal motion on each joint with different amplitude and frequency to capture a new dataset of the joints and the torques at each time step.

When the two criteria mentioned in Section 2.B.II are satisfied, $\boldsymbol{\theta_{final}}$ is equal to the actual functions of the differential equations of the robot, $\boldsymbol{\theta}_{\mathbf{actual}}$. The actual sparse vector $\xi_{iactual}$ in Equation 13 will be determined using the least square solution or least norm solution as, mentioned in Section 2.B.

$$\boldsymbol{\tau_i} = \boldsymbol{\theta}_{\mathbf{actual}}\,(Q)\,\xi_{iactual} \tag{13}$$

If the two criteria mentioned in Section 2.B are not satisfied, the application of SINDy is necessary to reduce $\boldsymbol{\theta_{final}}$ and get $\boldsymbol{\theta}_{\mathbf{actual}}$. Consequently, the sequentially threshold algorithm is used with the sparsity promoting parameter λ as presented in Equation 5 or 6 to solve Equation 11. As mentioned earlier in Section 2.B.II, there will be a torque model for every value of λ. The final models ($\boldsymbol{\theta}_{\mathbf{actual}}\,(Q)\ \&\xi_{iactual}$) are those with the minimum MSE.

III. RESULTS AND DISCUSSION

The proposed framework was applied to three case studies — Double Pendulum, Two-link of Kuka Robot-KR 4 AGILUS,

TABLE 2. Step II obtained minimum-error differential equations.

Link	Differential Equations
1	$\sin(\theta_2)\dot{\theta}_2^2 = 109.06\sin(\theta_1) + 36.1\sin(\theta_1+\theta_2) + 0.70915\ddot{\theta}_2 + 1.0017\cos(\theta_2)\ddot{\theta}_2 + 1.9991\cos(\theta_2)\ddot{\theta}_1 + 3.4201\ddot{\theta}_1 - 2.004\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2$ $\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 = 54.304\sin(\theta_1) + 18.069\sin(\theta_1+\theta_2) + 0.35448\ddot{\theta}_2 + 0.49938\cos(\theta_2)\ddot{\theta}_2 + 0.99761\cos(\theta_2)\ddot{\theta}_1 + 1.7048\ddot{\theta}_1 - 0.49797\sin(\theta_2)\dot{\theta}_2^2$
2	$\ddot{\theta}_1 = -0.99866\ddot{\theta}_2 - 51.143\sin(\theta_1+\theta_2) - 1.4069\cos(\theta_2)\ddot{\theta}_1 - 1.4065\sin(\theta_2)\dot{\theta}_1^2$ $\ddot{\theta}_2 = -1.0006\ddot{\theta}_1 - 51.226\sin(\theta_1+\theta_2) - 1.4094\cos(\theta_2)\ddot{\theta}_1 - 1.4085\sin(\theta_2)\dot{\theta}_1^2$ $\cos(\theta_2)\ddot{\theta}_1 = -0.70902\ddot{\theta}_1 - 0.70891\ddot{\theta}_2 - 36.339\sin(\theta_1+\theta_2) - 0.99817\sin(\theta_2)\dot{\theta}_1^2$ $\sin(\theta_2)\dot{\theta}_1^2 = -0.70914\ddot{\theta}_1 - 0.70879\ddot{\theta}_2 - 36.263\sin(\theta_1+\theta_2) - 0.99863\cos(\theta_2)\ddot{\theta}_1$

TABLE 3. Final differential equations of the system.

Link	Differential Equation	Approach
1	$\boldsymbol{\tau_1} = 3.741\sin(\theta_1) + 1.247\sin(\theta_1+\theta_2) + 0.02436\ddot{\theta}_2 + 0.03434\cos(\theta_2)\ddot{\theta}_2 + 0.06867\cos(\theta_2)\ddot{\theta}_1 + 0.1174\ddot{\theta}_1 - 0.03434\sin(\theta_2)\dot{\theta}_2^2 - 0.06867\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2$	*Proposed Methodology*
	$\boldsymbol{\tau_1} = -0.0335\sin(\theta_2)\dot{\theta}_2^2 + 3.6455\sin(\theta_1) + 1.2846\sin(\theta_1+\theta_2) + 0.0242\ddot{\theta}_2 + 0.0343\cos(\theta_2)\ddot{\theta}_2 + 0.0707\cos(\theta_2)\ddot{\theta}_1 + 0.1137\ddot{\theta}_1 - 0.0678\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2$	*Euler-Lagrange*
2	$\boldsymbol{\tau_2} = 0.0244\ddot{\theta}_1 + 0.0244\ddot{\theta}_2 + 1.2467\sin(\theta_1+\theta_2) + 0.0343\cos(\theta_2)\ddot{\theta}_1 + 0.0344\sin(\theta_2)\dot{\theta}_1^2$	*Proposed Methodology*
	$\boldsymbol{\tau_2} = 0.02436\ddot{\theta}_1 + 0.02436\ddot{\theta}_2 + 1.247\sin(\theta_1+\theta_2) + 0.03434\cos(\theta_2)\ddot{\theta}_1 + 0.03434\sin(\theta_2)\dot{\theta}_1^2$	*Euler-Lagrange*

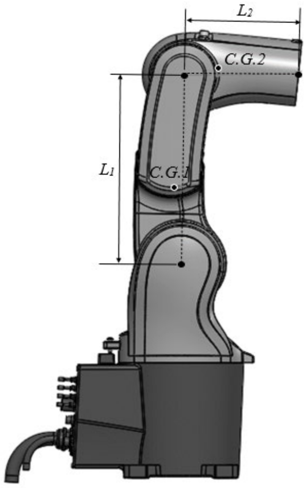

FIGURE 4. CAD model of the KUKA Robot: KR 4 AGILUS.

and Two-link of Stanford robot. The following is a discussion of the findings.

A. CASE STUDY-I

The schematic of the double pendulum is depicted in Fig.2. Table 1 summarizes the parameters of the system.

1) STEP I

The double pendulum CAD model was created in MATLAB, along with its SIMULINK model. The model will run for 10 seconds with a time step of 0.001 seconds to gather the training data ($\boldsymbol{q}$,$\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$). The data matrix is well conditioned, so Equation 5 is employed in the next step.

TABLE 4. Input parameters of case study II.

Parameter	Value
L_1	0.29 *m*
L_2	0.17 *m*
m_1	4.7171 *kg*
m_2	2.1414 *kg*
I_1 (Inertia of link 1) *kg.m*^{2} (Related to frame at C.G)	$\begin{bmatrix} 0.0624 & 0.0000 & -0.0011 \\ 0.0000 & 0.0546 & -0.0005 \\ -0.0011 & -0.0005 & 0.0171 \end{bmatrix}$
I_2 (Inertia of link 2) *kg.m*^{2} (Related to frame at C.G)	$\begin{bmatrix} 0.0034 & -0.0001 & 0.0003 \\ -0.0001 & 0.0104 & 0.0000 \\ 0.0003 & 0.0000 & 0.0102 \end{bmatrix}$
g (Gravity)	[0 -9.80665 0] ^{T} *m/s*^{2}

2) STEP II

Following the use of Equations 4, 5, 6, 7, 8, and 9, Table 2 displays the differential equations for the system that have the least MSE (Equation 9). Table 2 lists the functions that were chosen from the library, which may accurately or very closely represent the system.

As we can see from Table 2, there is no need to apply SINDy because those equations achieved the required criteria that were mentioned in Section 2.B.II

3) STEP III

A force $\boldsymbol{f_y} = \boldsymbol{3N}$ is applied to the end effector to compute the torque of the joints at each time step, as shown in Equation 12. The Least square solution is used to get the final equations (Table 3) that characterize the system. The resulting

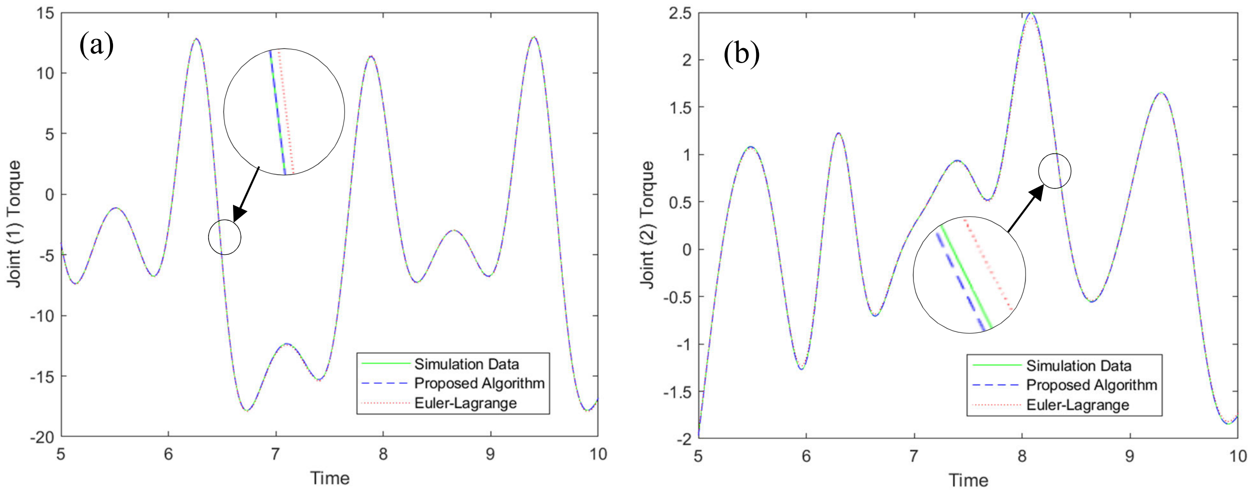

FIGURE 5. Comparison between joint torque calculated from simulated data, proposed algorithm and Euler Lagrange. (a) joint 1, (b) joint 2.

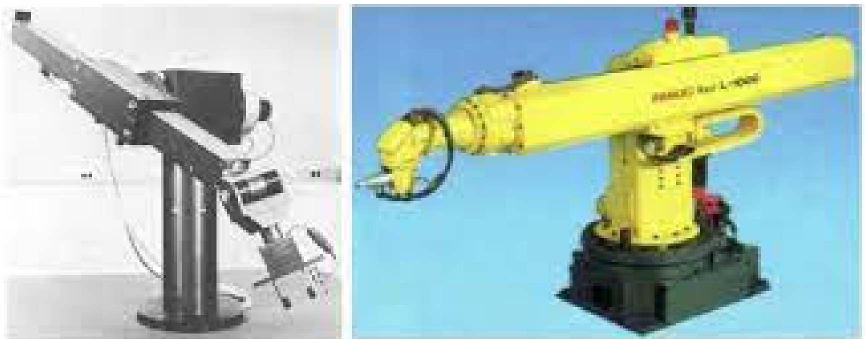

FIGURE 6. Stanford manipulator [33].

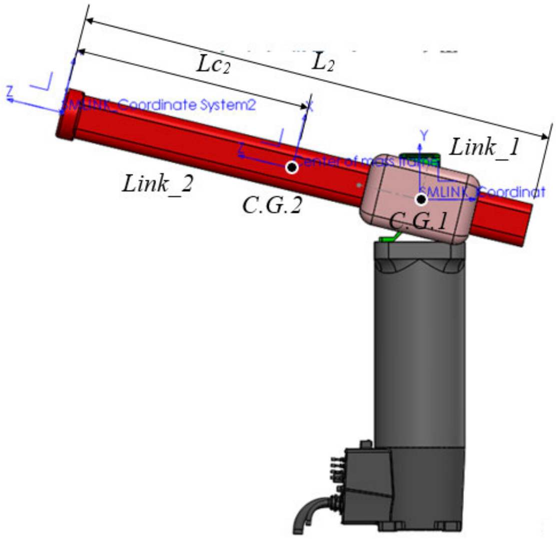

FIGURE 7. CAD model of Stanford manipulator.

differential equations and those derived by Euler-Lagrange method are very similar, as shown in Table 3.

B. CASE STUDY-II

The framework also was applied on two-link of KUKA KR 4 AGILUS [31] which is depicted in Fig.3. Table 4 shows the parameters of the Robot. The robot's CAD model is shown in Fig.4.

1) STEP I

The CAD model (Fig.4) was imported into MATLAB and the training data matrix $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ was created as mentioned in Section 1.B. The cond function in MATLAB was used to examine the condition of the data matrix. We found that, the data matrix is sensitive to the inverse computation (ill conditioned matrix). Despite, that the proposed framework was able to handle this situation.

2) STEP II

Following the same procedure as step II in case study I, the differential equations for the robot with the minimum MSE, were determined (Table 5).

Despite the fact that this robot has two revolute joints, as the double pendulum in Case Study I, the results were different. As seen in Table 5, the first two differential equations have identical terms, but the last two equations are not identical. Thus, the criteria mentioned in Section 2.B.II is not achieved, and the terms of the final differential equations cannot be determined. Therefore, the third stage is necessary to derive the final equations for the robot. All terms from

TABLE 5. Minimum-MSE differential equations of case study II.

Differential Equations
$\sin(\theta_2)\ddot{\theta}_1 = 468.42\cos(\theta_1+\theta_2) - 33.816\sin(\theta_1+\theta_2) + \cos(\theta_2)\dot{\theta}_1^2 + 13.852\sin(\theta_2)\dot{\theta}_1^2 + 13.852\cos(\theta_2)\ddot{\theta}_1 + 6.9991\ddot{\theta}_1 + 6.9991\ddot{\theta}_2$
$\cos(\theta_2)\dot{\theta}_1^2 = -468.42\cos(\theta_1+\theta_2) + 33.816\sin(\theta_1+\theta_2) + 0.002484\cos(\theta_2)\dot{\theta}_1^2 - 13.852\sin(\theta_2)\dot{\theta}_1^2 - 13.852\cos(\theta_2)\ddot{\theta}_1 + 0.002484\sin(\theta_2)\ddot{\theta}_1 - 6.9991\ddot{\theta}_1 - 6.9991\ddot{\theta}_2$
$\theta_1 = 1.1071\sin(\theta_1) - 0.30361{\theta_1}^2$
$\cos(\theta_2)\dot{\theta}_1\dot{\theta}_2 = 2528.5\cos(\theta_1) - 107.32\sin(\theta_1) - 0.5\cos(\theta_2)\dot{\theta}_1^2 - 0.5\cos(\theta_2)\dot{\theta}_2^2 - 6.926\sin(\theta_2)\dot{\theta}_1^2 - 6.926\sin(\dot{\theta}_2)\theta_2^2 + 6.926\cos(\theta_2)\ddot{\theta}_1 + 6.926\cos(\theta_2)\ddot{\theta}_2 - 0.5\sin(\theta_2)\ddot{\theta}_1 - 0.5\sin(\theta_2)\ddot{\theta}_2 + 65.88\ddot{\theta}_1 - 13.852\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2$

TABLE 6. Final differential equations of case study II.

Link	Differential Equation	Approach
1	$\boldsymbol{\tau_1} = 1.0701\cos(\theta_1+\theta_2) - 0.077254\sin(\theta_1+\theta_2) + 11.553\cos(\theta_1) - 0.49034\sin(\theta_1) - 0.0022845\cos(\theta_2)\dot{\theta}_2^2 - 0.031645\sin(\theta_2)\dot{\theta}_2^2 + 0.063291\cos(\theta_2)\ddot{\theta}_1 + 0.031645\cos(\theta_2)\ddot{\theta}_2 - 0.0045691\sin(\theta_2)\ddot{\theta}_1 - 0.0022845\sin(\theta_2)\ddot{\theta}_2 - 0.0045691\cos(\theta_2)\dot{\theta}_1\dot{\theta}_2 + 0.317\ddot{\theta}_1 + 0.01599\ddot{\theta}_2 - 0.063291\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2$	*Proposed Methodology*
	$\boldsymbol{\tau_1} = 1.05\cos(\theta_1+\theta_2) - 0.084\sin(\theta_1+\theta_2) + 11.641\cos(\theta_1) - 0.46259\sin(\theta_1) - 0.002484\cos(\theta_2)\dot{\theta}_2^2 - 0.03105\sin(\theta_2)\dot{\theta}_2^2 + 0.062101\cos(\theta_2)\ddot{\theta}_1 + 0.03105\cos(\theta_2)\ddot{\theta}_2 - 0.004968\sin(\theta_2)\ddot{\theta}_1 - 0.002484\sin(\theta_2)\ddot{\theta}_2 - 0.004968\cos(\theta_2)\dot{\theta}_1\dot{\theta}_2 + 0.31888\ddot{\theta}_1 + 0.015788\ddot{\theta}_2 - 0.062101\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2$	*Euler-Lagrange*
2	$\boldsymbol{\tau_2}1.0701\cos(\theta_1+\theta_2) - 0.077254\sin(\theta_1+\theta_2) + 0.0022845\cos(\theta_2)\dot{\theta}_1^2 + 0.031645\sin(\theta_2)\dot{\theta}_1^2 + 0.031645\cos(\theta_2)\ddot{\theta}_1 - 0.0022845\sin(\theta_2)\ddot{\theta}_1 + 0.015788\ddot{\theta}_1 + 0.01599\ddot{\theta}_2$	*Proposed Methodology*
	$\boldsymbol{\tau_2} = 1.05\cos(\theta_1+\theta_2) - 0.084\sin(\theta_1+\theta_2) + 0.002484\cos(\theta_2)\dot{\theta}_1^2 + 0.03105\sin(\theta_2)\dot{\theta}_1^2 + 0.03105\cos(\theta_2)\ddot{\theta}_1 - 0.002484\sin(\theta_2)\ddot{\theta}_1 + 0.015788\ddot{\theta}_1 + 0.015788\ddot{\theta}_2$	*Euler-Lagrange*

TABLE 7. Input parameters of case study III.

Parameter	Value
L_2	*1.02 m*
L_{c2}	*0.5 m*
m_1	*3.09808 kg*
m_2	*7.3077874 kg*
I_1 (Inertia of link 1) *kg.m*^{2} (Related to frame at C.G)	$\begin{bmatrix} 0.01426 & 0.00000 & 0.00000 \\ 0.00000 & 0.02066 & 0.00000 \\ 0.00000 & 0.00000 & 0.01961 \end{bmatrix}$
I_2 (Inertia of link 2) *kg.m*^{2} (Related to frame at C.G)	$\begin{bmatrix} 0.0492867 & 0.0000000 & 0.1554710 \\ 0.0000000 & 0.6445782 & 0.0000000 \\ 0.1554710 & 0.0000000 & 0.6028739 \end{bmatrix}$
g (Gravity)	$[0\ -9.80665\ 0]^T$ *m/s*^{2}

the four equations in Table 5 will be merged to create a new library.

3) STEP III

Following the procedures in Section 3.B.II, a two sinusoidal motion with different amplitude and frequency was applied to the joints. The data matrix $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}\ and\ \boldsymbol{\tau})$ was recorded. The SINDy algorithm was applied as stated in Section 3.B.II and the proposed framework was able to retrieve the true dynamic equations. The resulting differential equations and those derived by the Euler-Lagrange method are extremely comparable, as shown in Table 6 and Fig. 5.

C. CASE STUDY-III

The framework was applied to a two-link of Stanford robot, which is shown in Fig.6. The CAD model is shown in Fig.7 and its parameters are explained in Table 7. The same procedures of step I in the case of study I and case study II were applied to case study III. The condition of the data matrix was examined, and we found that it is an ill-conditioned matrix.

1) STEP II

The differential equations with the minimum MSE are shown in Table 8. The two criteria in Section 2.B.II were achieved in this case study. Thus, in step II, the SINDy algorithm is not needed and only regression is sufficient.

2) STEP III

As we can see from Table 9 and Fig.8, the resulting differential equations from regression and those derived by the Euler-Lagrange method are extremely comparable.

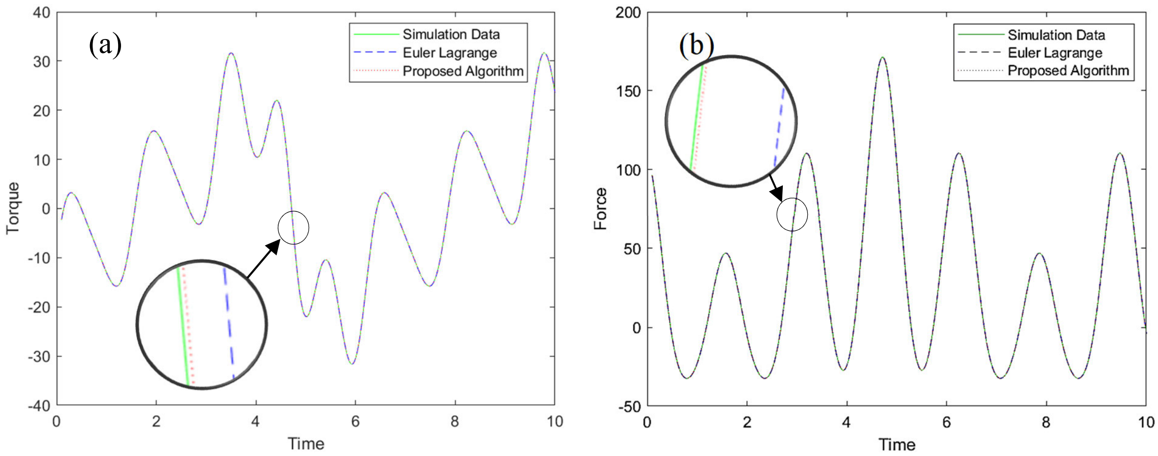

FIGURE 8. Comparison between joint torque/force calculated from simulated data, proposed algorithm and Euler Lagrange. (a) joint 1, (b) joint 2.

TABLE 8. Minimum-error differential equations of case study III.

Link	Differential Equations
1	$\ddot{\theta} = 3.3086\ddot{\theta}d - 16.223\sin(\theta) + 3.3086\dot{\theta}\dot{d} - 5.1678d^2\ddot{\theta} + 50.679\sin(\theta)d - 10.336\dot{\theta}d\dot{d}$
	$\ddot{\theta}d = 0.30224\ddot{\theta} + 4.9033\sin(\theta) - \dot{\theta}\dot{d} + 1.5619d^2\ddot{\theta} - 15.317\sin(\theta)d + 3.1239\dot{\theta}d\dot{d}$
	$d^2\ddot{\theta} = 0.64023\ddot{\theta}d - 3.1393\sin(\theta) - 0.19351\ddot{\theta} + 0.64023\dot{\theta}\dot{d} + 9.8067\sin(\theta)d - 2.0\dot{\theta}d\dot{d}$
2	$\ddot{d} = d(\dot{\theta})^2 - 9.8067\cos(\theta) - 0.32012(\dot{\theta})^2$
	$d(\dot{\theta})^2 = 0.32012(\dot{\theta})^2 + \ddot{d} + 9.8067\cos(\theta)$

TABLE 9. Final differential equations of case study III.

Link	Differential Equation	Approach
1	$\boldsymbol{\tau_1} = 7.30779d^2\ddot{\theta} - 4.67698d\ddot{\theta} + 1.413556\ddot{\theta} + 14.6156\dot{\theta}d\dot{d} - .67698\dot{\theta}\dot{d} - 71.6649\sin(\theta)d + 22.9328\sin(\theta)$	*Proposed Methodology*
	$\boldsymbol{\tau_1} = 7.30779d^2\ddot{\theta} - 4.67698d\ddot{\theta} + 1.413556\ddot{\theta} + 14.6156\dot{\theta}d\dot{d} - 4.67698\dot{\theta}\dot{d} - 71.6649\sin(\theta)d + 22.9328\sin(\theta)$	*Euler-Lagrange*
2	$\boldsymbol{\tau_2} = 71.665\cos(\theta) + 2.3385\left(\dot{\theta}\right)^2 - 7.3078d\left(\dot{\theta}\right)^2 + 7.3078\ddot{d}$	*Proposed Methodology*
	$\boldsymbol{\tau_2} = 71.665\cos(\theta) + 2.3385\left(\dot{\theta}\right)^2 - 7.3078d\left(\dot{\theta}\right)^2 + 7.3078\ddot{d}$	*Euler-Lagrange*

IV. CONCLUSION

This paper introduced a framework for data-driven dynamic modeling of serial manipulators based on the SINDy algorithm. The framework consists of three steps: library generation, library reduction, and extraction of the robot's differential equation of motion. The framework was proposed to deal with the ill-conditioned data matrix that arises in case of serial robots with complex geometry. The framework was applied to three different case studies —Double pendulum, two-link KUKA robot, two-link Stanford robot. When the framework was applied to a robot with a more complex geometry than that of the double pendulum, it accurately obtained the terms of the differential equations. Due to robot's complexity, the data matrix obtained from the library generation step is ill-conditioned. The functions that accurately represent the dynamics are selected through the library reduction process, which employs the SINDy -PI method and the sequential threshold technique with the least norm solution. The sequential threshold technique with the least norm solution showed better performance than the least square error solution and lasso regression that were used in previous studies. The force applied to the end effector or the application of a trajectory in joint space in the final step allows the robot's torque equations to be accurately obtained. The suggested framework demonstrated that, contrary to earlier work, it is not necessary to choose a special trajectory to accurately capture all the proper terms of the robot's dynamic equations. The result of the suggested framework is compared to the results from the Euler-Lagrange method, which validate the results of the proposed framework. In the future, we plan to apply the methodology to more complex industrial robot configurations with higher degrees of freedom. Also, experimental data will be employed rather than simulation data to investigate the effect of the measurements' noise on the results of the framework.

V. ABBREVIATIONS

SINDy:	Sparse Identification of Non-Linear Dynamics
SINDy-PI:	Parallel Implicit Sparse Identification of Nonlinear Dynamics
SINDy-c:	Sparse cation of Non-Linear Dynamics with control
SINDy-PDE:	SINDy For Partial Differential Equations

REFERENCES

[1] L. Tsai, *Robot Analysis and Design: The Mechanics of Serial and Parallel Manipulators*. New York, NY, USA: Wiley, 1999. [Online]. Available: http://www.loc.gov/catdir/bios/wiley042/98036463.html

[2] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ, USA: Wiley, 2020.

[3] D. Dai, "A systematic procedure for applying robot dynamics-based control," Ph.D. dissertation, Dept. Electron. Comput. Eng., Hong Kong Univ. Sci. Technol., Hong Kong, 2016. [Online]. Available: https://repository.hkust.edu.hk/ir/Record/1783.1-86630

[4] J. Chen, M. Zhang, Z. Yang, and L. Xia, "A robust data-driven approach for dynamics model identification in trajectory planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Prague, Czech Republic, Sep. 2021, pp. 7104–7111, doi: 10.1109/IROS51168.2021.9635979.

[5] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 474, no. 2219, Nov. 2018, Art. no. 20180335, doi: 10.1098/rspa.2018.0335.

[6] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015, doi: 10.1126/science.aaa8415.

[7] V. Marx, "Biology: The big challenges of big data," *Nature*, vol. 498, no. 7453, pp. 255–260, 2013, doi: 10.1038/498255a.

[8] H. K. Chu and M. Hayashibe, "Discovering interpretable dynamics by sparsity promotion on energy and the Lagrangian," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2154–2160, Apr. 2020, doi: 10.1109/LRA.2020.2970626.

[9] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," 2017, *arXiv:1708.00588*.

[10] M. Raissi and G. E. Karniadakis, "Machine learning of linear differential equations using Gaussian processes," 2017, *arXiv:1701.02440*.

[11] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, "Predicting catastrophes in nonlinear dynamical systems by compressive sensing," 2011, *arXiv:1105.0462*.

[12] G. Gouesbet and C. Letellier, "Global vector-field reconstruction by using a multivariate polynomial L_2 approximation on nets," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 49, no. 6, pp. 4955–4972, Jun. 1994, doi: 10.1103/physreve.49.4955.

[13] L. Wei-Dong, K. F. Ren, S. Meunier-Guttin-Cluzel, and G. Gouesbet, "Global vector-field reconstruction of nonlinear dynamical systems from a time series with SVD method and validation with Lyapunov exponents," *Chin. Phys.*, vol. 12, no. 12, pp. 1366–1373, Dec. 2003, doi: 10.1088/1009-1963/12/12/005.

[14] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.

[15] R. Gonzàlez-García, R. Rico-Martínez, and I. G. Kevrekidis, "Identification of distributed parameter systems: A neural net based approach," *Comput. Chem. Eng.*, vol. 22, pp. S965–S968, Mar. 1998, doi: 10.1016/S0098-1354(98)00191-4.

[16] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse identification of nonlinear dynamics with control (SINDYc)," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016, doi: 10.1016/j.ifacol.2016.10.249.

[17] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," 2016, *arXiv:1609.06401*.

[18] M. Stender, S. Oberst, and N. Hoffmann, "Recovery of differential equations from impulse response time series data for model identification and feature extraction," *Vibration*, vol. 2, no. 1, pp. 25–46, Jan. 2019, doi: 10.3390/vibration2010002.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," 2016, *arXiv:1602.04938*.

[20] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, Apr. 2009.

[21] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016, doi: 10.1073/pnas.1517384113.

[22] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Inferring biological networks by sparse identification of nonlinear dynamics," 2016, *arXiv:1605.08368*.

[23] M. Quade, M. Abel, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for rapid model recovery," 2018, *arXiv:1803.00894*.

[24] K. Kaheman, J. N. Kutz, and S. L. Brunton, "SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics," *Proc. Roy. Soc. A. Math., Phys. Eng. Sci.*, vol. 476, no. 2242, Oct. 2020, Art. no. 20200279, doi: 10.1098/rspa.2020.0279.

[25] K. Kaheman, S. L. Brunton, and J. N. Kutz, "Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data," 2020, *arXiv:2009.08810*.

[26] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

[27] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 473, no. 2204, Aug. 2017, Art. no. 20170009, doi: 10.1098/rspa.2017.0009.

[28] K. P. Burnham and D. R. Anderson, *Model Selection and Multi-Model Inferencs: A Practical Information-Theoretic Approach*, 2nd ed. New York, NY, USA: Springer, 2010.

[29] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*, E. Parzen, K. Tanabe, and G. Kitagawa, Eds. New York, NY, USA: Springer, 1998, pp. 199–213.

[30] H. Akaike, "A new look at the statistical model identification," in *Selected Papers of Hirotugu Akaike*, E. Parzen, K. Tanabe, and G. Kitagawa, Eds. New York, NY, USA: Springer, 1998, pp. 215–222.

[31] KUKA. *KUKA Robot-KR 4 AGILUS*. Accessed: Sep. 2022. [Online]. Available: https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/kr-4-agilus

[32] GrabCad. *CAD Model of Kuka Robot: KR 4 AGILUS*. Accessed: Aug. 2022. [Online]. Available: https://grabcad.com/library/kuka-robotic-kr-4-agilus-1

[33] G. Palacios and O. Alvarez, "The generalized architecture of the spherical serial manipulator," *Adv. Robot. Autom.*, vol. 5, no. 2, p. 38, 2016, doi: 10.4172/2168-9695.1000148.

MOHAMED OMAR received the B.S. and M.S. degrees in production and mechanical design engineering from Mansoura University, Mansoura, Egypt, in 2014 and 2018, respectively. He is currently pursuing the Ph.D. degree in mechanical engineering with the Harbin Institute of Technology, Harbin, China.

From 2014 to 2022, he was a Research and Teaching Assistant with the Faculty of Engineering, Mansoura University.

RUIFENG LI received the B.E., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1996, respectively. He is currently a Professor and the Vice Director with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His research interests include artificial intelligence and robotics. He is a member of the Chinese Association for Artificial Intelligence and the President of the Heilongjiang Province Institute of Robotics.

AHMED ASKER received the M.Sc. and Ph.D. degrees from the Egypt-Japan University of Science and Technology, Egypt, in 2014 and 2017, respectively. He joined the University of Leeds as a Research Fellow in rehabilitation robotics, from 2020 to 2022. He is currently a Lecturer with the Production and Mechanical Design Engineering Department, Mansoura University, Egypt, and an Adjunct Assistant Professor at Nile University, Egypt. His research interests include assistive robotics, parallel manipulators, and machine learning.