Real-Time Physically-Accurate Simulation of Robotic Snap Connection Process

Minji Lee, Jeongmin Lee, Jaemin Yoon and Dongjun Lee[†]

***Abstract*— We propose a novel real-time physically-accurate simulation framework for the snap connection process. For this, we first notice the peculiarities of the process, namely, small/smooth deformation, stiff connector and segmented contact. We then design our simulation to fully exploit these peculiarities by adopting the following strategies: 1) the technique of passive midpoint integration (PMI [1]), which allows for stable simulation of arbitrarily light/stiff system by enforcing discrete-time passivity; 2) linear finite element method (FEM [2]) modeling, which is adequate to deal with the small snap connector deformation while providing much faster speed as compared to nonlinear FEM; 3) segmentation of the snap connector FEM model and solving of each segment individually with their coupling analytically eliminated, thereby, further speeding up the simulation; 4) balanced model reduction (BMR [3]) to further reduce the dimension of each segment purely analytically without any *prior* experiment or simulation; and 5) parallelized data-driven collision detection, which turns out to further significantly speed up our simulation. Experimentally-verified simulations are also performed to show the efficacy of our proposed simulation framework.**

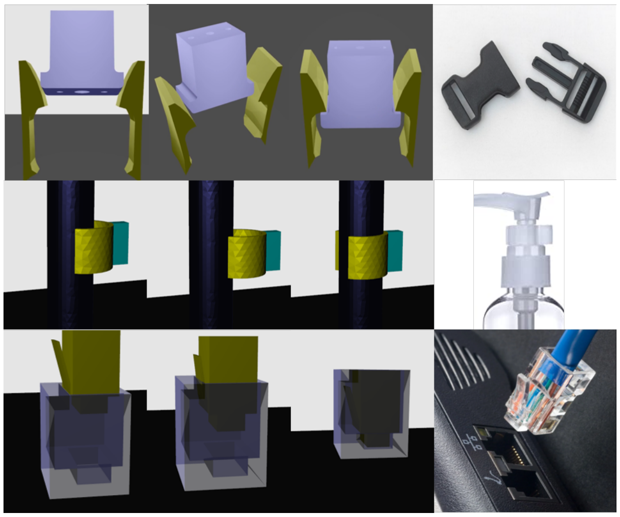


Fig. 1: Simulation snapshots of various real-world snap connection processes

I. INTRODUCTION

With the recent advancements of robotics technologies, many attempts have been made to bring the robots into real-world applications, including manufacturing/assembly and household operations (e.g., [4]). One of processes frequently recurring for those applications is the snap connection process, where a component is forcefully pushed against another component with a complementary geometry in such a way that, by exploiting the slight deformation between them, the two components can be slid and "snapped" together to join with each other. This process is not only important for robotic manufacturing/assembly (e.g., plastic switch box assembly, inserting LAN cable, etc.), but also for household robotics (e.g., tidying objects with snap-belts, placing dishwasher stopper, etc.) - see Fig. 1.

Fast and accurate simulation of this snap connection process would then be useful for the development of its model-based robot control strategy, and also so would be the case for the development of data-driven reinforcement learning (RL [5]) strategy, which recently receives lots of attention, yet, whose sim-to-real (e.g., [6], [7]) typically requires a vast number of simulation data. Its real-time simulation would even allow for the haptic rendering of the snap connection in VR (virtual reality [8]) or the development of shared-autonomy RL strategy [9] for the snap connection.

In this paper, we develop a novel *real-time physically-accurate* simulation framework for this snap connection process. For this, we first notice the following peculiarities of the process: 1) the connection deformation itself is not so large; 2) the deformed connector shape is not of so high spatial frequency; 3) the connector is fairly stiff to exhibit fast snap closing motion (with "clicking"); and 4) typically not all the parts of the connectors are under the contact at the same time. We then render our simulation framework to exploit these peculiarities s.t.,: 1) we derive our simulation based on the technique of passive midpoint integration (PMI [1]), which allows for stable simulation of arbitrarily light/stiff system by enforcing discrete-time passivity; 2) we choose linear finite element method (FEM [2]) instead of nonlinear FEM [10], as it is enough to deal with the small deformation of snap connection while providing much faster speed; 3) we split the snap connector FEM model into few segments and solve them individually with their coupling analytically eliminated (see Sec. III-A), thereby, speeding up the simulation time; 4) we adopt the balanced model reduction (BMR [3]) to further reduce the dimension of each segmented model purely analytically without any *prior* experiment or simulation as required in the data-driven model reduction methods (e.g., [11]); and 5) we devise a parallelized data-driven collision detection module between the connector surfaces in the form of multilayer perceptron (MLP [12]) and with the help of graphics processing unit

---

The authors are with the Department of Mechanical Engineering, IAMD and IER, Seoul National University, Seoul, Republic of Korea, 08826. {mingg8,ljmlgh,yjm5181,djlee}@snu.ac.kr. Corresponding author: Dongjun Lee.

Research supported by the Industrial Strategic Technology Development Program (20001045) of the Ministry of Trade, Industry & Energy (MOTIE) of Korea.

(GPU), which turns out to further significantly speed up the simulation.

There are numerous results in the fields of robotics and computer graphics on the fast simulation of deformable objects with multi-point contacts based on nonlinear FEM, and their state-of-the-art results may be categorized by optimization-based approaches (e.g., [13]–[15]) or inversion-based approaches (e.g., [11]). However, all of them (e.g., [11], [13]–[15]) typically rely on an implicit way to address nonlinear functions (e.g., stiffness) in their iteration or integration, which is well-known to increase the damping effect in the simulation (e.g., [1]), thus, making it impossible for them to properly simulate the fast snap closing motion (clicking) - a signature behavior of many snap connection processes. Of course, an exact solver of the nonlinear functions can remove this damping effect, yet, at the cost of giving up the real-time speed of the simulation. Further, the data-driven hyper-reduction method, as adopted in [11], necessitates *prior* experiments/simulations to build the basis for the simulation. In contrast to these, our proposed framework can real-time simulate the fast snap closing motion (thanks to the adoption of PMI) while not requiring any *prior* experiments/simulations for the model reduction (thanks to the adoption of BMR). We also believe that linear FEM is enough for the snap connection process simulation, and, to our knowledge, our proposed framework is the very first result of the real-time snap connection process simulation fully utilizing the peculiarities of the process.

The rest of the paper is organized as follows. Sec. II introduces the snap connector setting and other basic theoretical tools. Sec. III derives the FEM model of the socket connector, its segmentation and model reduction via BMR. Sec. IV presents the parallelized collision detection and the contact solving procedure. Performance of our simulation is then presented/analyzed in Sec. V, followed by some concluding remarks in Sec. VI.

II. PRELIMINARY

A. Snap Connector Setting

We consider a snap connection, typically consists of a rigid plug connector and a deformable socket connector, and plug connector is forcefully squeezed into the complementary geometry of the socket connector. Once the plug connector is completely inserted, the structural features of the socket connector prevent it from being plugged out. The socket connector is modeled with FEM, which is discretized by tetrahedrons. Since the deformation of each prong barely affects each other, we separate them into two different FEM models without interaction - see Fig.2. The dynamics of each prong of the socket connector can then be formulated as

$$M_t\ddot{x}_t + C_t\dot{x}_t + K_t(x)x_t = f_{t,ext} \tag{1}$$

where $x_t \in \mathbb{R}^{3n_t}$ is the stacked deformation of the FEM nodes, $M, D, K \in \mathbb{R}^{3n_t\times 3n_t}$ are the mass, damping and stiffness matrices respectively and $f_{t,ext} \in \mathbb{R}^{3n_t}$ is the external force applied to each node (i.e. via the interaction with the plug connector) when n_t denotes the number of the nodes of each model.

In (1), K_t is dependent on x, so it should be calculated at every time step. However, since small deformations can be assumed in the snap connector, its behavior can be approximated by a linearized FEM model around the configuration x_0, which is then given by a linear time invariant (LTI) dynamic system with a constant stiffness matrix. With the bottom of the socket connector fixed to the floor, we can reduce its dynamics by removing the columns and the rows of M_t, D_t and K_t about those fixed (bottom) nodes. The socket connector dynamics is then given by:

$$M\ddot{x} + D\dot{x} + Kx = f_{ext} \tag{2}$$

where $x \in \mathbb{R}^{3n}$ is the displacement of the reduced FEM nodes, $M, D, K \in \mathbb{R}^{3n\times 3n}$ are the reduced matrices of M_t, D_t, K_t, $f_{ext} \in \mathbb{R}^{3n}$ is the external force on the nodes rather than the fixed nodes and n is the number of the reduced nodes.

On the other hand, the plug connector does not deform much, so it can be assumed as a rigid object. Then, the dynamics of the plug connector can be written as $M_{pl}\dot{V}_{pl} = \mathcal{F}_{ext}$, where $M_{pl} := \text{diag}\left([m_{pl}I_{3\times 3}, I_{pl}]\right) \in \mathbb{R}^{6\times 6}$, with $m_{pl}, I_{pl} \in \mathbb{R}$ being the mass and inertia of the plug connector, and $V_{pl} := [v_{pl}; w_{pl}] \in \mathbb{R}^6$, with $v_{pl}, w_{pl} \in \mathbb{R}^3$ being the linear and angular velocity respectively, and $\mathcal{F}_{ext} \in \mathbb{R}^6$ is the external wrench exerted on the plug connector.

B. Passive Midpoint Integration

Passive midpoint integration (PMI) is a dynamics integration method that guarantees discrete time passivity [1]. We adopt this PMI here, since it allows us to real-time simulate any arbitrarily stiff/light system behavior (e.g., fast snap connection motion), while also generally improves the stability of the simulation. A linearized version of the FEM dynamics (2) is discretized via PMI as:

$$M\frac{v_{k+1}-v_k}{T} + D\hat{v}_k + K\hat{x}_k = f_k, \quad \hat{v}_k := \frac{v_{k+1}+v_k}{2} = \frac{x_{k+1}-x_k}{T} \tag{3}$$

where $\star_k := \star(t_k)$, $v_k := \dot{x}_k$ is linear velocity of the nodes and T denotes the step size of time. Let us denote by $\hat{\star}_k$ the representative of $\star_k$ for the duration $t \in [T_k, T_{k+1}]$. Then the dynamics (3) can be organized as

$$\hat{v}_k = \hat{M}^{-1}f_k + v_{f,k} \tag{4}$$

where $\hat{M} := \frac{2M}{T} + D + K\frac{T}{2}$ and $v_{f,k} := \hat{M}^{-1}\left(\frac{2}{T}Mv_k - Kx_k\right)$.

If the plug connector is controlled via admittance controller, PMI-discretized dynamics of the plug connector can be written as

$$\hat{V}_{pl,k} = \hat{M}_{pl}^{-1}\mathcal{F}_{ext} = \hat{M}_{pl}^{-1}\left(\mathcal{F}_{pl,k} + \mathcal{F}_{c,k}\right) + V_{pl,f,k} \tag{5}$$

where $V_{pl,f,k} := \hat{M}_{pl}^{-1}\left(\frac{2}{T}M_{ad}V_{pl,k} - K_{ad}X_{pl,k}\right)$, $\hat{M}_{pl} := \frac{2}{T}M_{ad} + D_{ad}$, $M_{ad}, D_{ad}, K_{ad} \in \mathbb{R}^6$ are the chosen mass, damper and stiffness matrices in admittance dynamics and $\mathcal{F}_c, \mathcal{F}_{pl}$ are the contact wrench and actuation wrench (exerted by the robot arm) respectively.

Thanks to the PMI, the dynamics (4) (and also (5)) satisfies discrete-time energetic passivity in the sense that: $\forall N \geq 0$,

$$\sum_{k=0}^{N} f_{ext}^T \hat{v}_k T \geq E_N - E_0 \tag{6}$$

where $E_k := \frac{1}{2}\dot{x}_k^T M \dot{x}_k + \frac{1}{2} x_k^T K x_k$ is the total energy. This passivity property of PMI allows us to choose arbitrarily large K or small M. We use PMI discretization for our FEM dynamics to ensure the discrete time passivity, so that the snap connector with high Young's modulus can be simulated stably without artificial damping.

C. Balanced Model Reduction

Balanced model reduction (BMR) is an analytic projection-based model order reduction method [16]. It projects the state into a subspace with smaller dimension which is well balanced between observability and controllabilty. Unlike data-driven methods such as proper orthogonal decomposition (POD) [11], it does not require data, which means that prior experiments or simulations are not demanded. In this paper, we adopt this BMR to further reduce the dimension of each segmented FEM model - see Sec.III-B.

The FEM dynamics (2) can be transformed into first-order state-space form with state $X := [\dot{x}; x] \in \mathbb{R}^{6n}$, where n denotes the number of the FEM nodes:

$$\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \dot{X} = \begin{bmatrix} -D & -K \\ I & 0 \end{bmatrix} X + \begin{bmatrix} B \\ 0 \end{bmatrix} u \tag{7}$$

where $u \in \mathbb{R}^{3n_b}$ is the external force, $B \in \mathbb{R}^{3n \times 3n_b}$, is the selection matrix of the input nodes that can be subjected to external forces, when n_b being the number of input nodes, and $x = [0_{3n\times 3n} \;\; I_{3n\times 3n}]X$ is an output of the system for the BMR. In order to enforce the second-order structure of the FEM model dynamics, we adopt the second-order BMR result of [17] then, the controllability gramian and the observability gramian, $W_c, W_o \in \mathbb{R}^{3n\times 3n}$ for the state x can be computed as described in [16], and the transformation to the reduced space can be written as

$$x =: \bar{T} z, \quad \bar{T} := R^T U \Lambda^{-\frac{1}{2}} \tag{8}$$

where $R^T R := W_{c,x}$, $U\Lambda^2 U^T := R W_{o,x} R^T$, $U^T U = U U^T = I$, and $\Lambda \in \mathbb{R}^{n\times n}$ is composed of Hankel singular values of the balanced gramian of $\bar{z}$. The BMR result can then be achieved by choosing

$$z := [\bar{z}_1; \cdots; \bar{z}_{n^r}] = P\bar{z} \in \mathbb{R}^{n^r} \tag{9}$$

where $P = \left[I_{n^r\times n^r} \;\; 0_{n^r\times(3n-n^r)}\right]$ is the selection matrix of the highest to the n^r-th highest Hankel singular values.

III. SEGMENTATION WITH MODEL REDUCTION

We develop a segmentation technique with BMR to accelerate the simulation. Linear FEM dynamics have the advantage that the inverse of the matrices for dynamics update can be precalculated. With the precalculated matrices, the state variables can be updated via matrix multiplication. Since the computational complexity of matrix multiplication is about $\mathcal{O}(n^{2.4})$, we can effectively speed up this stage by splitting the dynamics into several dynamics with a lower degree of freedom. Moreover, BMR is applied to the segmentation technique to further reduce the dimension and reduce the computational cost. Recall that these are consistent with the peculiarities of the snap connection process as stated in Sec. I.

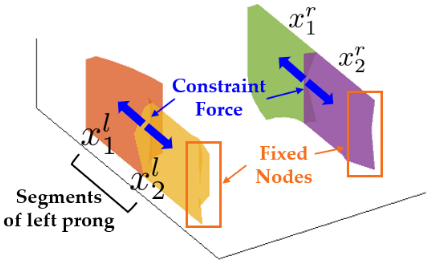


Fig. 2: Segmentation of the socket connector FEM model

A. Segmentation of Socket Connector

We propose an implicit spring-based segmentation framework. The FEM model is geometrically divided into segments, which are connected with a strong spring. The socket connector FEM model (2) is segmented into the nodes with possible contact (e.g., nodes on the surface) and the nodes with no contact (e.g., internal nodes). Although the socket connector FEM model (2) can be divided into a number of segments, here, for brevity, we give the derivation only for the two-part segmentation. Define $x_1 := [x_{11}; x_{12}; \cdots; x_{1n_1}]^T \in \mathbb{R}^{3n_1}, x_2 := [x_{21}; x_{22}; \cdots; x_{2n_2}]^T \in \mathbb{R}^{3n_2}$, where x_1, x_2 denote the nodes of the two segments where x_1 being the nodes with possible contacts, while x_2 not. Then, the dynamics of each segment can be written as

$$M_i \ddot{x}_i + D_i \dot{x}_i + K_i x_i = H_i^T \lambda_i + f_{i,c}, \quad i = 1,2 \tag{10}$$

where $\lambda_i \in \mathbb{R}^{3n_c}$ is the constraint force (to be designed below) of the i-th segment to ensure the joining between the two segments and $f_{i,c}$ is the contact force, $H_i \in \mathbb{R}^{3n_c \times 3n_i}$ is the selection matrix of the coupling nodes among the nodes in the i-th segment and n_c is the number of the coupling nodes. Since there is no contact on second segment, the contact force of the second segment is zero: $f_{2,c,k} = 0$.

Then, following the PMI expression, (10) can be discretized at the k-th time step:

$$\hat{v}_{i,k} = \hat{M}_i^{-1}\left(H_i^T \lambda_{i,k} + f_{i,c,k}\right) + v_{i,f,k} \tag{11}$$

where $v_{i,f,k} := \hat{M}_i^{-1}\left(\frac{2M_i}{T} v_{i,k} - K_i x_{i,k}\right)$ and $\hat{M}_i := \frac{2M_i}{T} + D_i + K_i \frac{T}{2}$. We also design the constraint force λ_i as a strong PMI-based spring-damper connection s.t.,

$$\lambda_{1,k} := -d_c \hat{v}_{e,k} - k_c \hat{x}_{e,k}, \quad \lambda_{2,k} = -\lambda_{1,k} \tag{12}$$

$$\hat{v}_{e,k} := H_1 \hat{v}_{1,k} - H_2 \hat{v}_{2,k}, \quad \hat{x}_{e,k} := H_1 \hat{x}_{1,k} - H_2 \hat{x}_{2,k} \tag{13}$$

where $x_{e,k}, v_{e,k} \in \mathbb{R}^{3n_c}$ denotes the position error and the velocity error at the coupling nodes respectively, and $k_c, d_c \in \mathbb{R}$ can be interpreted as PD gain for constraint stabilization. Thanks to our adoption of PMI for the simulation, we can choose arbitrarily large d_c, k_c for the (virtually-rigid) segmental coupling (12) without losing stability nor real-timeness, even with zero damping (i.e., $d_c = 0$ in the equation (12), $C_i = 0$ in the equation (10)) or BMR - see Prop. 1.

The segmented FEM model (11), however, contains the coupling force $\lambda_{i,k}$ with other segmented model, thus, necessitates us to solve dynamics of both segments simultaneously, implying that the simulation speed-up would be unlikely. To circumvent this necessity, here, we present a way to analytically eliminate this coupling $\lambda_{i,k}$, thereby, speeding up the simulation. More precisely, by substituting $\lambda_{2,k}$ (12) into the equation about $\hat{v}_{2,k}$ (11), $\hat{v}_{2,k}$ can be written about $\hat{v}_{1,k}$. Then by substituting this $\hat{v}_{2,k}$ into $\lambda_{1,k}$(12) with the obtained relation, $\lambda_{1,k}$ can be written only with $\hat{v}_{1,k}$. Then finally, by substituting $\lambda_{1,k}$ into equation (11), $\hat{v}_{1,k}$ can be written as a linear mapping of the contact force $f_{1,c,k}$ with the coupling $\lambda_{1,k}$ now eliminated:

$$\hat{v}_{1,k} = A f_{1,c,k} + v_{1,f,k} \tag{14}$$

Here, since the mapping matrix $A \in \mathbb{R}^{3n_1 \times 3n_1}$ is a time-invariant constant matrix, it can be pre-computed. As the large-dimensional state update equation is split into two equations with smaller dimensions via segmentation, parallelization is made possible and the computation time is reduced.

This spring-based segmentation is equivalent to the original FEM model if the constraints are strictly satisfied. Since our system is passive even when the spring coefficient between coupling nodes k_c are large, we can effectively ensure the tight connectivity by increasing k_c, which makes the solution close to the original problem.

B. Segmentation with Balanced Model Reduction

Performing BMR for each segment further reduces the state dimensions, thereby speeding up the computation. For this BMR, we need to choose the input nodes, i.e., nodes that can be subjected to external forces. This reduction can handle the contact force by including the nodes with a possibility of contact in the input nodes. These input nodes include the contact nodes (i.e., where the contact with the plug connector might occur, such as nodes on surfaces) and the coupling nodes (i.e., where the constraint force λ_i can be exerted).

Let $T_i \in \mathbb{R}^{3n_i \times 3n_i}$ be the transformation matrix to the balanced subspace, and $P_i \in \mathbb{R}^{n_i^r \times 3n_i}$ be the selection matrix of the modes with the n_i^r highest Hankel singular values of each i-th object. The full-order state $x_i \in \mathbb{R}^{3n_i}$ is projected to the reduced state $z_i \in \mathbb{R}^{n_i^r}$, and it can be described by $z_i = P_i T_i^{-1} x_i$. Also, the reconstruction of the full-order state can be written as $x_i = T_i P_i^T z_i$. With these relations, the state update equation (11) can be rewritten for z_i:

$$Q_i^T \hat{M}_i Q_i \dot{\hat{z}}_{i,k} = Q_i^T b_{i,k} + Q_i^T \lambda_{i,k} + Q_i^T f_{i,c,k}, \quad i = 1,2 \tag{15}$$

where $Q_i := T_i P_i^T, b_{i,k} = \frac{2M_i}{T} v_{i,k} - K_i x_{i,k}$. Also, the constraint force $\lambda_{i,k}$ (12) can be expressed as same as (12) with

$$\hat{x}'_{e,k} := H_1 Q_1 \hat{z}_{1,k} - H_2 Q_2 \hat{z}_{2,k}, \quad \hat{v}'_{e,k} := H_1 Q_1 \dot{\hat{z}}_{1,k} - H_2 Q_2 \dot{\hat{z}}_{2,k} \tag{16}$$

where v'_e, x'_e is the velocity and position error at the coupling nodes which is constructed in $z-$space. Following the steps outlined above, $\dot{\hat{z}}_{1,k}$ and $\dot{\hat{z}}_{2,k}$ can be written likewise as:

$$\dot{\hat{z}}_{2,k} = B_1 \dot{\hat{z}}_{1,k} + \dot{z}_{2,f,k} \tag{17}$$

$$\dot{\hat{z}}_{1,k} = B_2 f_{1,c,k} + \dot{z}_{1,f,k} \tag{18}$$

where $B_1 \in \mathbb{R}^{n_2^r \times n_1^r}$ and $B_2 \in \mathbb{R}^{n_2^r \times 3n_1}$ are constant matrices, which can be precalculated. In addition to the advantage of being split into dynamics equations of small dimensions that can be parallelized, the dimensions of the state variables have further decreased from $3n_i$ to n_i^r, which improves the computational speed.

Our segmentation with BMR method strictly satisfies discrete time passivity without any artificial damping as proved in Proposition 1.

Proposition 1. The segmentation method with BMR satisfies discrete time passivity (i.e. (6) with $E_k = \Sigma_i(\frac{1}{2} v_{i,k}^T M v_{i,k} + \frac{1}{2} x_{i,k}^T K x_{i,k}) + \frac{1}{2} x_{e,k}^T k_C x_{e,k})$ even without artificial damping (i.e., $d_c = 0$ in the equation (16), $C_i = 0$ in the equation (10)).

Proof. The PMI-discretized dynamics without any damping, with coupling force $\lambda_{i,k}$ substituted with (12), (16) can be written as follows:

$$\begin{aligned} &M_1 Q_1 \tfrac{\dot{z}_{1,k+1} - \dot{z}_{1,k}}{T} + K_1 Q_1 \hat{z}_{1,k} \\ &= f_1 - k_c H_1^T \left(H_1 Q_1 \hat{z}_{1,k} - H_2 Q_2 \hat{z}_{2,k}\right) \end{aligned} \tag{19}$$

$$\begin{aligned} &M_2 Q_2 \tfrac{\dot{z}_{2,k+1} - \dot{z}_{1,k}}{T} + K_2 Q_2 \hat{z}_{2,k} \\ &= -k_c H_2^T \left(H_2 Q_2 \hat{z}_{2,k} - H_1 Q_1 \hat{z}_{1,k}\right) \end{aligned} \tag{20}$$

By multiplying $T \dot{\hat{z}}_{1,k}^T Q_1^T$ and $T \dot{\hat{z}}_{2,k} Q_2^T$ to the left side of (19) and (20) respectively, equation about energy in z-space can be deduced as:

$$\begin{aligned} &\Sigma_i \tfrac{1}{2} (\dot{z}_{i,k+1}^T Q_i^T M_i Q_i \dot{z}_{i,k+1} - \dot{z}_{i,k}^T Q_i^T M_i Q_i \\ &\quad \dot{z}_{i,k} + z_{i,k+1}^T Q_i^T K_i Q_i z_{i,k+1} - z_{i,k}^T Q_i^T K_i Q_i z_{i,k}) \\ &= -\tfrac{1}{2} k_c \left((x'_{e,k+1})^T x'_{e,k+1} - (x'_{e,k})^T x'_{e,k} \right) + T \dot{\hat{z}}_{1,k}^T Q_1 f_1 \end{aligned} \tag{21}$$

With the relation $x_i = Q_i z_i$, the equation (21) in z-space can be transformed into x-space and be simplified as:

$$\begin{aligned} &\tfrac{1}{2} (v_{i,N}^T M_i v_{i,N}^T - v_{i,0}^T M_i v_{i,0} + x_{i,N}^T K_i x_{i,N} - x_{i,0}^T K_i x_{i,0} \\ &+ k_c (x_{e,N})^T x_{e,N} - k_c (x_{e,0})^T x_{e,0}) = T \hat{v}_{1,k} f_1 \end{aligned} \tag{22}$$

As (22) indicates that the work done by contact force is exactly the same with the total energy (i.e., E_k) change, it can be concluded that the passivity is satisfied. ☐

IV. CONTACT HANDLING

A. MLP-based Collision Detection

Collision detection between objects modeled by meshes involves heavy computation, especially when the number of the elements is large and the shape changes. To reduce the computation time at this stage, we develop a multilayer perceptron (MLP) based collision detection technique. Although this collision detection is data-driven, it can be obtained offline with computer-aided design (CAD) data, so no prior experiments or simulations are required.

For this, let us first define a signed distance function [18] of the plug connector, which determines the distance of a

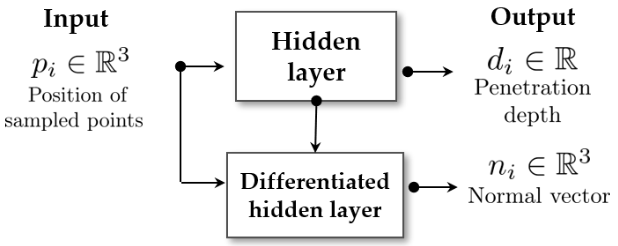


Fig. 3: Block diagram that represents the overall structure of the collision detection process

input point p from the surface of the plug connector. It has positive values for the points inside the surface, zero for the points on the surface, and negative values for the points outside the surface. If the distance function of the plug connector is obtained, we can check whether or not collision has occurred by simply checking the sign of the function value of the points on the surface of the socket connector. We approximate the distance function to MLP so that the point-mesh collision detection algorithm can be simplified into a neural network traversal process consisting of simple calculations.

The MLP is trained with numerous randomly generated points and their distance function values, which can be computed as the minimum signed distance value from the triangle on the surface of the plug connector to the point. The network has total four layers including input and output layer. The activation function for the input layer and the hidden layers is ReLU function and tanh function is used for the output layer.

Using this MLP-based point-mesh(A) collision detection, mesh(A)-mesh(B) collision detection can be performed by checking whether numerous points on mesh B are outside or inside the other mesh A. At every time step, a number of points $\bar{p} = [p_1; \cdots ; p_N]$ are obtained by interpolating FEM nodes of the socket connector (i.e., $x_{1,k} \in \mathbb{R}^{3n_1}$) and transforming the points into the plug connector coordinate. Then these transformed points pass through MLP, and the distance function value of each point $d = [d_1, \cdots , d_N]$ can be achieved. With the sign of the values, the contact points can be chosen.

Moreover, since the normal vectors can be interpreted as the gradients of the signed distance functions, the normal vectors of the contact points can be achieved by passing through the differentiated MLP. The whole process of detecting the contact points and obtaining normal vectors is presented in Fig. 3.

As we generate numerous points in the process of collision detection, several points that are close to each other can be detected as contact points. These points might induce numerical instability while calculating contact force [19] and slow down the calculation. We adopt k-means clustering [20] to classify the obtained contact points into a predetermined number of clusters, and use the point which has the largest penetration in each cluster as a representative point.

B. Contact Solver

The next step is to calculate the contact force on the obtained contact points. We solve the contact force based on the maximal dissipation principle [21] with the obtained contact points and normal vectors.

Let us first define the contact coordinate of the *j*-th contact point which is formed with the normal vector and the tangential plane of the triangular mesh that contains the contact point. The relative velocity between the two connectors in the contact coordinate is required for the contact solver. The contact solving process is in the following order: 1) compute the velocity of the contact points on the socket connector; 2) compute the velocity of the contact points on the plug connector; then 3) formulate the contact optimization problem with the relative velocity in the contact coordinate.

1) Velocity of the contact points on the socket connector: By using linear interpolation along each triangular surface mesh with barycentric coordinates, the velocity of the contact point can be written as a linear combination of the velocity of the nodes:

$$v^c_{p,k} = J_s v_{1,k} \tag{23}$$

where, superscript c means that the property is described in the contact coordinate, $v^c_{p,k} \in \mathbb{R}^{3m}$ is the stacked velocity of the contact points on the socket connector, m is the number of the contact points, and $J_s \in \mathbb{R}^{3m \times 3n_1}$ maps the FEM nodes into the contact coordinate. Likewise, the force of the contact points are interpolated with the nodal force as $f^c_{1,k} = J_s^T f_{1,c,k}$. Finally, with the relation proposed in (14) can be transformed to the contact coordinate as

$$\hat{v}^c_{p,k} = J_s A J_s^T f^c_k + J_s v_{1,f,k} \tag{24}$$

2) Velocity of the contact points on the plug connector: Similarly, the stacked velocity of the contact points on the plug connector $\hat{v}^c_{pl,k} \in \mathbb{R}^{3m}$ also can be expressed as linear mapping of the velocity of the plug connector:

$$v^c_{pl,k} = J_{pl} V_{pl,k} \tag{25}$$

where, $J_{pl} \in \mathbb{R}^{3m \times 6}$ is the linear mapping between the velocity of plug connector and the velocity of contact points, and $V_{pl,k} = [v_{pl}; w_{pl}] \in \mathbb{R}^6$. Then by transforming (5) into contact coordinate with (25), velocity-force relation of plug connector in contact coordinate can be written as:

$$\hat{v}^c_{pl,k} = -J_{pl} \hat{M}^{-1}_{pl} J^T_{pl} f^c_k + J^T \left(\hat{M}^{-1}_{pl,k} \mathcal{F}_{pl} + V_{pl,f,k} \right) \tag{26}$$

Then, the relative velocity of contact points on the socket connector about the plug connector $\hat{v}^c_{r,k} = \hat{v}^c_{1,k} - \hat{v}^c_{pl,k}$ ((24) - (26)) can be computed as

$$\hat{v}^c_{r,k} = C f^c_k + v_{r,f,k} \tag{27}$$

where $C := J_s A J_s^T + J_{pl} \hat{M}^{-1}_{pl} J^T_{pl}$ and $v_{r,f,k} := J_s v_{1,f,k} - J_{pl} \left(\hat{M}^{-1}_{pl} \mathcal{F}_{pl} + V_{pl,f,k} \right)$. The delassus operator C should be calculated at every time step, but the dimension is $3m$ by $3m$, where m is the number of clustered points.

3) Calculating Contact Force: With the relation described in (27), we compute the contact force that satisfies Coulomb's friction and Signorini condition. If the Delassus operator has good matrix condition (e.g., mass-spring systems with diagonal mass and stiffness matrices), projective Gauss-Seidel (PGS) [22] method will show good convergence. However in the case of the snap connection process, since the contact force is applied from two symmetrical directions, the condition of the Delassus operator is bad and the PGS method does not converge well. As an alternative, we use maximal dissipation algorithm [21] which finds the contact force that maximizes the dissipation. The contact optimization problem can then be written as

$$\begin{aligned} \min_{f_k^c} \quad & \left(\hat{v}^c_{r,k}\right)^T f_k^{c(t)} = \left(C f_k^c + v_{r,f,k}\right)^T f_k^{c(t)} \\ \text{subject to} \quad & \mu \| f_k^{c(n)} \| \leq \| f_k^{c(t)} \| \\ & v_\varepsilon^{c(n)} + \frac{d}{T} = 0 \end{aligned} \tag{28}$$

where superscript $(n), (t)$ mean normal and tangential component of the vector respectively, $v^c_\varepsilon := v^c_{r,k+1} + \varepsilon v^c_{r,k} = 2\hat{v}^c_{r,k} + (\varepsilon - 1) v_{r,k}$, ε is the coefficient of restitution, and μ is the coefficient of friction. In addition, $d = [d_1; \cdots; d_m]$ and d_j denotes the distance value of the j-th contact point calculated in Sec. IV-A. The physical meaning of the first constraint is the Coulomb's friction cone and the second constraint is the Signorini condition in velocity level compensated for penetrations that have already occurred before being detected.

Hwangbo et al. [23] proposed a method to substitute a single contact problem into a simple line search problem and used the bisection method to solve it. By solving each single contact problem and updating the total contact force with successive over relaxation (SOR), it efficiently solves the multi-contact problem. Our contact formulation is found to have the same optimization formula, that is finding the point with the minimal Mahalanobis norm to the objective point in a feasible set of ellipses. Therefore, we adopt the bisection method for each single contact and SOR method to compute the multi-contact force.

V. Simulation Analysis

A. Stability With High Stiffness

To show the better stability property of our linear FEM-based simulator as compared to the nonlinear FEM approaches, we performed the comparison between our simulator and the corotational FEM [24], which is widely used for FEM simulation, both with high Young's modulus. Two models are integrated with PMI. As proved in Proposition 1, our framework can stably simulate the snap connection behavior even when the damping term is removed (i.e., $D_i = 0$ in the equation (10)). However, although the framework is analytically stable, numerical error may take place, which implies that small damping is required. Unlike our framework, if the material is very stiff (i.e., high Young's modulus), typically linearized dynamics of the nonlinear FEM can diverge due to the dynamics linearization error regardless of the integration method.

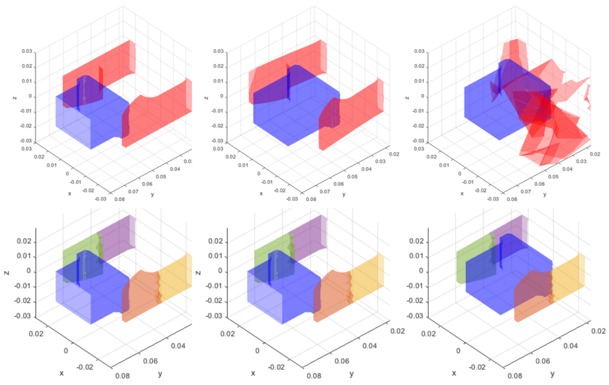


Fig. 4: Snapshots of the simulation results: nonlinear FEM (with geometric nonlinearity and linear material) (top) and proposed framework (bottom).

Snapshots of the simulation results of this nonlinear FEM and our proposed simulation model are presented in Fig. 4. Our method is simulated without any artificial damping and the corotational FEM is simulated with small Reyleigh damping - $D = \alpha M + \beta K$, $\alpha = 1.0 \times 10^{-4}$, $\beta = 1.0 \times 1e-5$. We can see that our proposed simulator can stably simulate the snap connection process with high stiffness ($E = 164.7$ Pa) even without damping, whereas the nonlinear FEM simulator just diverges.

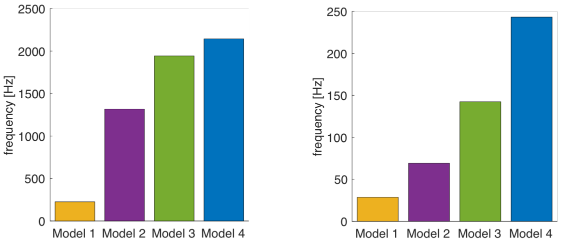
B. Computation Time Analysis

	Original FEM	Segmentation	BMR
Coarse Mesh	663	165 84	6 25
Fine Mesh	1266	876 465	6 96

TABLE I: The dimensions (i.e., number of the nodes×3) of: 1) the original snap connector FEM models with coarse or fine meshes; 2) their segmented models; and 3) their reduced models via BMR.

Here, we analyze the computation time of our proposed simulator and how effective each adopted idea is to speed up the simulation. We implement the simulator with C++ and parallelize the collision detection with CUDA developed by NVIDIA. TABLE. I shows the dimensions (i.e., number of the nodes×3) on each stage for each finely and coarsely divided FEM model. When the original mesh is discretized more finely, the dimension of the state is reduced by greater amount.

With the reduced dimension and the GPU-based collision detection, the simulation frequency (i.e. number of time steps calculated in 1 second) was increased by 8.5-9.5 times faster for different sizes of meshes. Fig. 5 shows the frequency - average number of time steps calculated per second - of the simulation of each model. Exploiting data-driven collision detection, segmentation and BMR has enhanced the simulation frequency. In the case of the coarse mesh, data-driven collision detection affected the simulation time remarkably, whereas, in the case of the fine mesh, the effect of segmentation and reduction was huge.



(a) Coarse Mesh (b) Fine Mesh

Fig. 5: Simulation frequency comparison between 4 models - Model 1: original FEM model; Model 2: original FEM model with GPU collision detection; Model 3: Model 2 with segmentation; and Model 4: Model 3 with BMR.

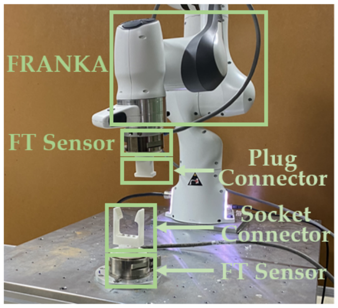


Fig. 6: Experiment setup consists of FRANKA robot manipulator, ATI gamma force/torque sensors, and snap connectors

C. Simulation Result with Experimental Data

To verify the framework, we implement the framework with C++ and compare the experimental results with the simulations. We print snap connector models with a 3D printer with ABS filament.

1) Parameter Identification: We identify the material parameters of the socket connector and the friction coefficient between the two connectors to match our simulation framework to the real world.

We exert force on the socket connector and observe the deformation of each node of the FEM model. The force is applied by a robot arm (FRANKA EMIKA Panda) and two ATI gamma force/torque sensors (FT sensor) are attached below the socket connector and between the robot arm and the plug connector. Deformation of each snap connector node, represented as x_{DIC}, is measured by open source digital image correlation (DIC) algorithm [25]. We move the plug connector slowly (i.e. quasi-static), which makes justifiable to simplify the FEM dynamics (2) as $Kx = f_{ext}$. Then, we can compute the nodal displacement by $x = K^{-1}f_{ext}$ when the input force is known, remarking that the stiffness matrix K is nonsingular. The normal vector of each mesh is required to calculate the force from the measured force with the FT sensor. To obtain the normal vector of the deformed socket connector surface, x_{DIC} is used. The objective function is defined as:

$$\arg\min_{E} \| x_{DIC} - K^{-1} f_{ext} \|$$

Here we adopt the known value of the Poisson's ratio of ABS ($\nu = 0.35$) to parameterize stiffness matrix K only with the Young's modulus E, which makes the optimization problem more efficient. The solution is obtained by MATLAB fmincon function with interior-point method. As a result, we achieve $E = 1.6476 \times 10^8$ Pa which is plausible under the known range of the material property. Note that the parameter identification is irrelevant with the proposed methods - segmentation, model reduction and MLP-based detection.

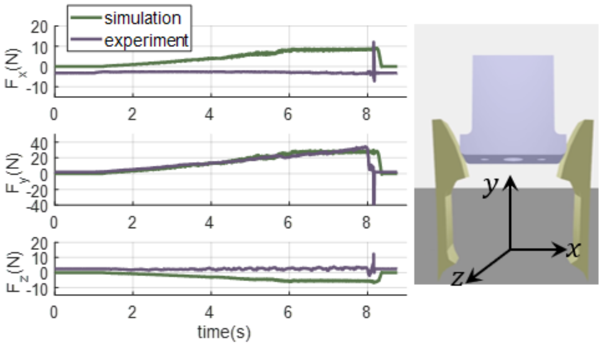


Fig. 7: Comparison of the force measured with FT sensor and that calculated from proposed simulation.

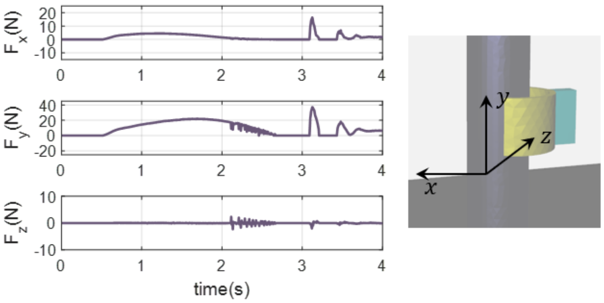
To construct the contact model, the friction coefficient between the two connectors should be identified. The socket connector, with a pendulum on the top, is placed on the plug connector, and the force is exerted horizontally. Then we measure the force at the moment when the socket connector starts to move. With the measured force, the friction coefficient can be identified as $\mu = f_h/W_s$, where f_h is the measured horizontal force and W_s is the weight of the socket connector. Ultimately, we achieved $\mu = 0.163$.

2) Comparison with Experiment: To compare our simulation with the experiment, we first conduct snap connections using the admittance control of the robot. Then the simulation is performed using the same control input (i.e. desired position of the plug connector) with the experiment. Here, time step is used as 1 ms. Since the friction in robot arm joints, which is difficult to model in simulation, leads to unavoidable differences between simulation and experiment, the admittance gain was set to the value that best fits the experiment.

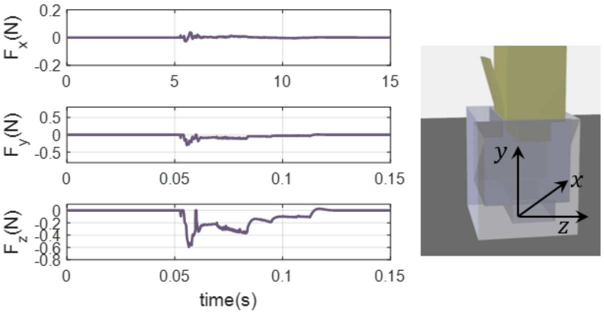
With the FT sensor attached below the socket connector, we measure the force exerted to the floor and compare it with the calculated force in the simulation. The force applied to the fixed nodes can be directly calculated in simulation using the discretized FEM dynamics about the fixed nodes as:

$$\hat{v}_{a,k} = \hat{M}_{fx} f_{fx} + v_{a,f,k} \tag{29}$$

where $\hat{M}_{fx} := \frac{2}{T}M_{fx} + D_{fx} + \frac{T}{2}K_{fx}$, $v_{a,f,k} := \hat{M}_{fx}^{-1}\left(\frac{2}{T}M_{fx}v_{a,k} - K_{fx}x_{a,k}\right)$, and M_{fx}, D_{fx}, K_{fx} are mass, damping and stiffness matrices consists of the removed rows from M_t, D_t and K_t. Fig. 7 illustrates the comparison of the measured force from the experiment and that calculated from proposed simulation. The root mean square error of the norm of the force was 4.936 N, where the maximum force was 48.159 N.



(a) plastic leakage prevention cap-like geometry



(b) LAN cable connector-like geometry

Fig. 8: Calculated force exerted on the floor for two different snap connectors.

D. Simulation of Different Type Connectors

Our framework was tested with connectors with different geometries and materials to validate the robustness. We modeled two connectors which imitates LAN cable connector and plastic leakage prevention cap. The material parameter of the former one is $E = 50.0$ MPa, $\nu = 0.35$ and latter one is $E = 100$ MPa, $\nu = 0.1$. As shown in Fig. 8, the proposed framework stably simulates other snap connectors with different geometries and material parameters.

VI. CONCLUSION

We present a real-time snap connection framework that matches real-world physics. For this, segmentation and BMR are conducted to reduce the dimension of the states. Moreover, data-driven collision detection using GPU parallelization is proposed to speed up the detection. As a result, we achieve the frequency of the simulation up to 2.2 kHz. The devised framework is then verified through comparison with the experiment and it is confirmed that the accuracy was not lost in the process of reduction.

Some future research directions include: 1) using different frames for each segment so that large variations can be handled while maintaining the advantages of the framework; 2) adopting the segmentation framework in nonlinear FEM simulations for more general objects; 3) make our simulation algorithm in an open source packet.

REFERENCES

[1] M. Kim, Y. Lee, Y. Lee, and D. Lee. Haptic rendering and interactive simulation using passive midpoint integration. *International Journal of Robotics Research*, 36(12):1341–1362, 2017.

[2] T. JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[3] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.

[4] ABB Robotics. Assembly application of electrical sockets with yumi-irb14000.

[5] Andrychowicz OM., Baker B., Chociej M., and et al. Learning dexterous in-hand manipulation. *International Journal of Robotics Research*, 39(1):3–20, 2020.

[6] J. Matas, S. James, and A. J. Davison. Sim-to-real reinforcement learning for deformable object manipulation. *Proceedings of Machine Learning Research (CoRL)*, 87:734–743–23, 2018.

[7] D. Son, H. Yang, and D. Lee. Sim-to-real transfer of bolting tasks with tight tolerance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[8] Y. Lee, M. Kim, Y. Lee, J. Kwon, Y. Park, and D. Lee. Wearable finger tracking and cutaneous haptic interface with soft sensors for multi-fingered virtual manipulation. *IEEE/ASME Transactions on Mechatronics*, 24(1):67–77, 2018.

[9] S. Reddy, A. Dragan, and S. Levine. Shared autonomy via deep reinforcement learning. 2018.

[10] T. Belytschko, W. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2013.

[11] O. Goury and C. Duriez. Fast, generic, and reliable control and simulation of soft robots using model order reduction. *IEEE Transactions on Robotics*, 34(6):1565–1576, 2018.

[12] S. Haykin. *Neural Networks and Learning Machines*. Pearson Education India, 2010.

[13] M. Ly, J. Jouve, L. Boissieux, and F. Bertails-Descoubes. Projective dynamics with dry frictional contact. *ACM Transactions on Graphics*, 39(4):57–1, 2020.

[14] C. Brandt, E. Eisemann, and K. Hildebrandt. Hyper-reduced projective dynamics. *ACM Transactions on Graphics*, 37(4):1–13, 2018.

[15] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and V. Makoviychuk. Non-smooth newton methods for deformable multi-body dynamics. *ACM Transactions on Graphics*, 38(5):1–20, 2019.

[16] J. Yoon, I. Hong, and D. Lee. Passive model reduction and switching for fast soft object simulation with intermittent contacts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6963–6970, 2019.

[17] D. Meyer and S. Srinivasan. Balancing and model reduction for second-order form linear systems. *IEEE Transactions on Automatic Control*, 41(11):1632–1644, 1996.

[18] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M-P Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, et al. Collision detection for deformable objects. In *Computer graphics forum*, volume 24, pages 61–81, 2005.

[19] A. Rocchi, B. Ames, Z. Li, and K. Hauser. Stable simulation of underactuated compliant hands. In *IEEE International Conference on Robotics and Automation*, pages 4938–4944, 2016.

[20] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[21] T. Preclik, S. Eibl, and U. Rüde. The maximum dissipation principle in rigid-body dynamics with inelastic impacts. *Computational Mechanics*, 62(1):81–96, 2018.

[22] P. Horak and J. Trinkle. On the similarities and differences among contact models in robot simulation. *IEEE Robotics and Automation Letters*, 4(2):493–499, 2019.

[23] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018.

[24] M. Müller and M. H. Gross. Interactive virtual materials. In *Graphics interface*, volume 2004, pages 239–246, 2004.

[25] J. Blaber, B. Adair, and A. Antoniou. Ncorr: open-source 2d digital image correlation matlab software. *Experimental Mechanics*, 55(6):1105–1122, 2015.