MATLAB EXPO 2022

Autonomous Co-working Robot Development with Multimodal Control based on Model-Based Design

Kazuki Ono

May, 2022

KYOCERA

Agenda

1. Introduction
2. Software Development Kit(SDK) for Robot-Hand Control System
3. Overview of Robot-Hand Development using Model-Based Design(MBD) and MATLAB® Tools
4. Construction of Controllable Multimodal Robot System
5. Impression and Conclusion

1. Introduction

1.1 Initiatives for Multimodal Autonomous Co-working Robot System

Autonomous Co-working Robot With Multimodal Control System

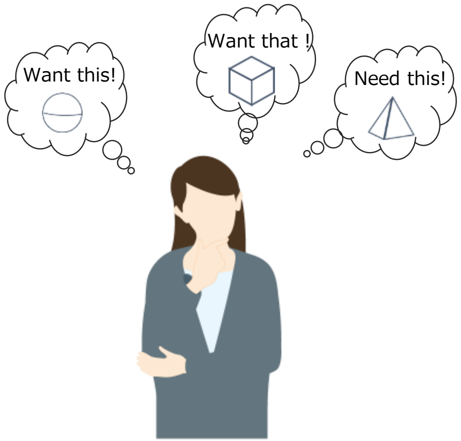
An adaptive reactive system which capable of multimodal interactions and changes dynamically in response to external environment (i.e. self-modification feature)

1.2 Background of Co-working Robot Applications

Social Challenges:

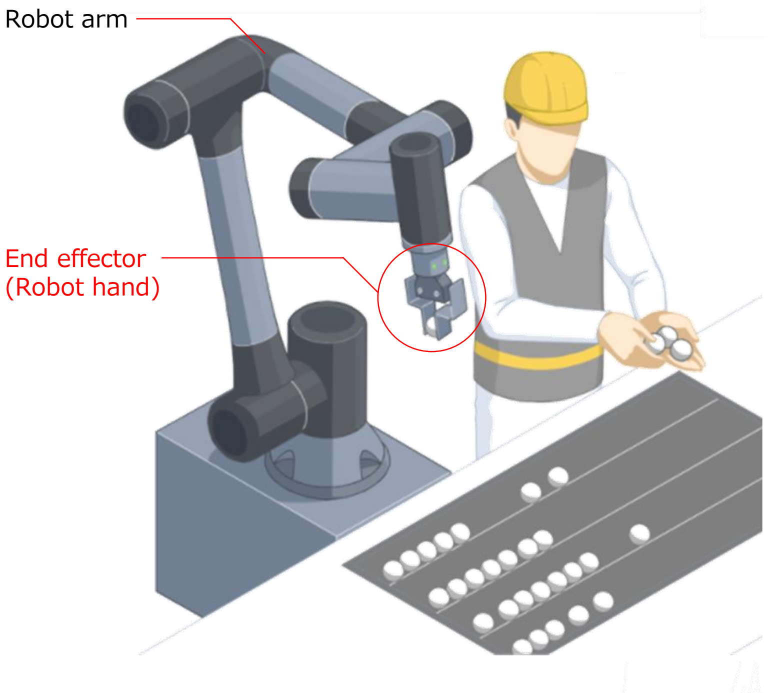
+ Shrinking workforce caused by aging and declining population

+ Diverse customer demands



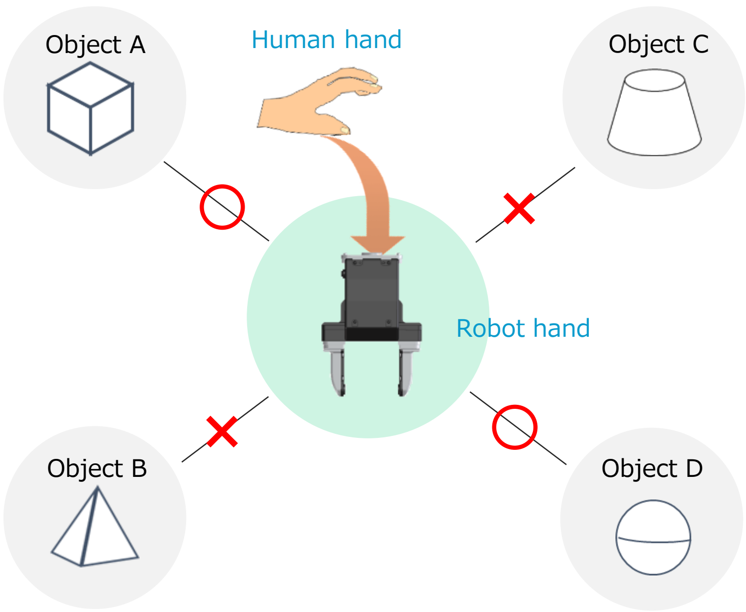
Solution

Co-working Robot with Multimodal Control System

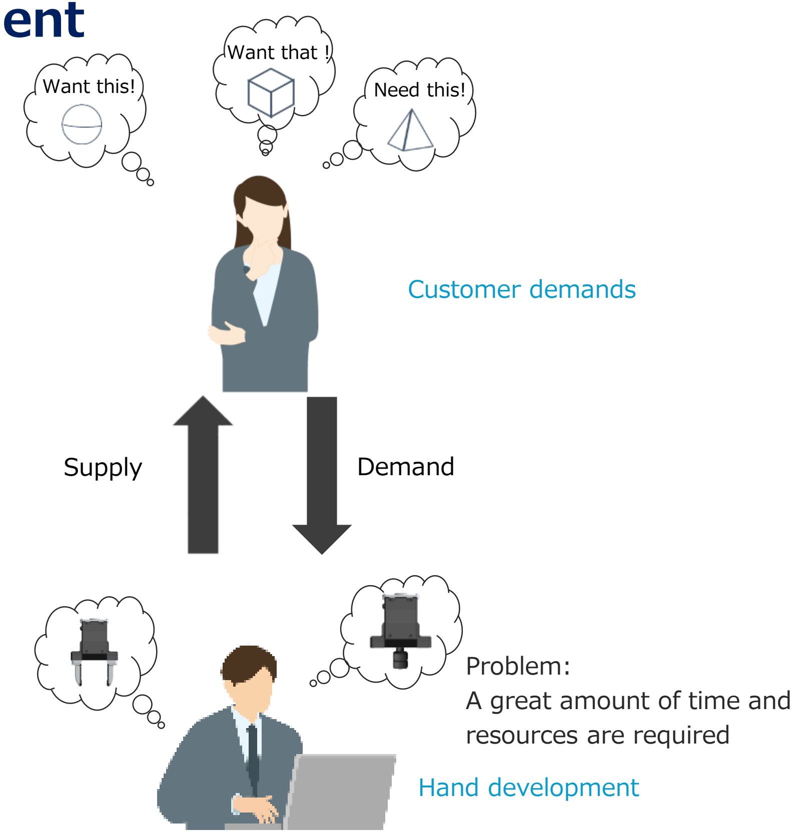


Choosing an end effector is critical to manage multiple and diverse customer demands

1.3 Background of Robot Hand Development

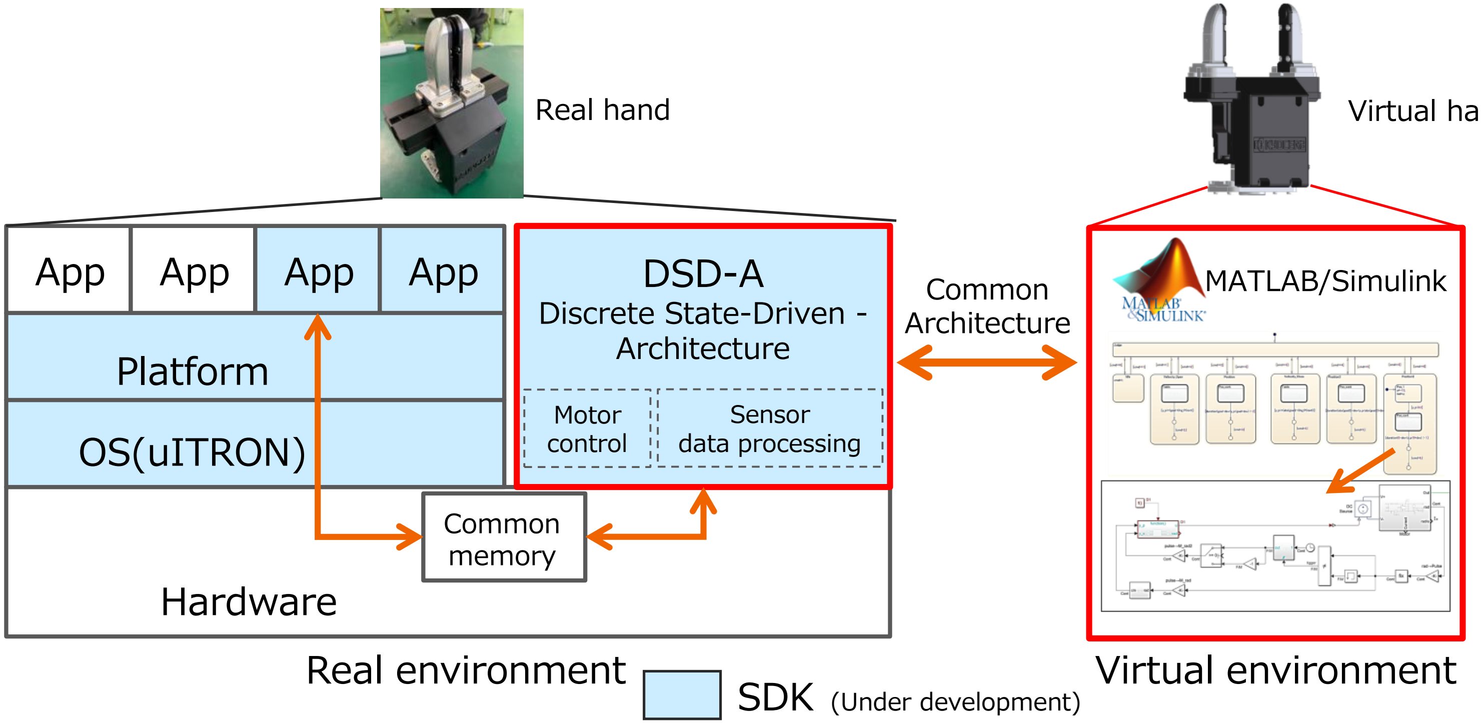


No practical application of a general-purpose hand that can replace human hand to perform any task. Generally, robot hand development is required for each specific task.



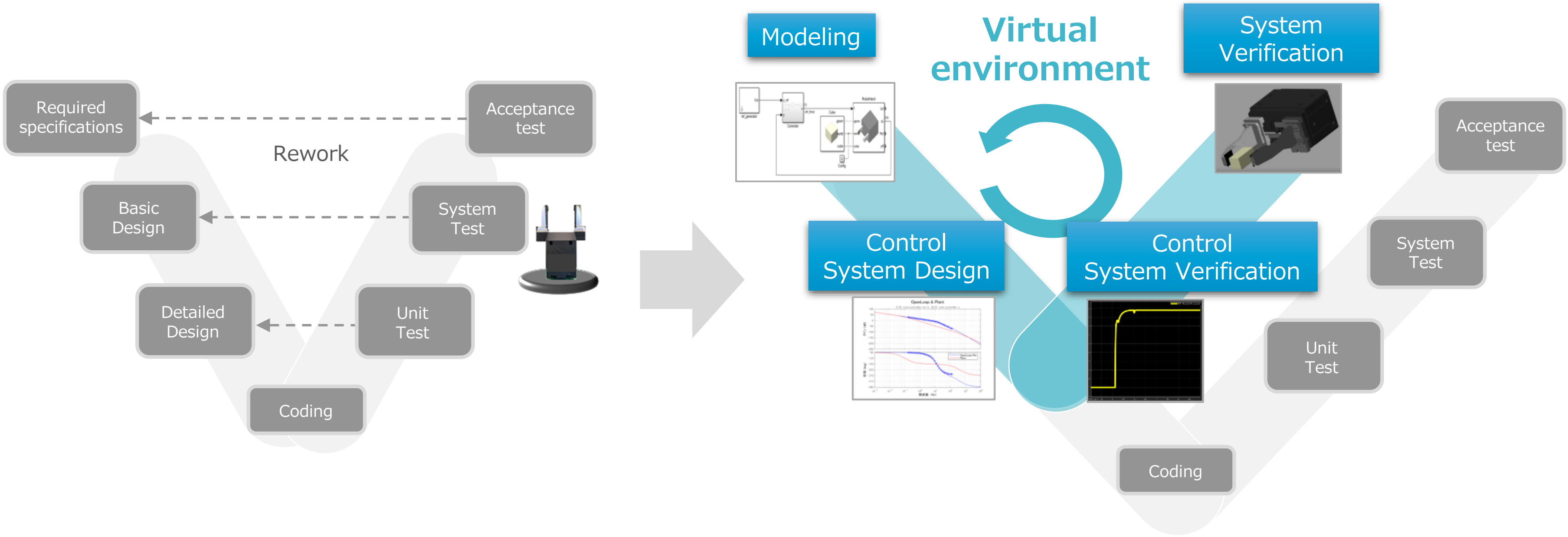
Rapid robot hand development system is necessary to satisfy diverse customer demands

1.4 Our proposal : Software Dev. Kit (SDK) for Robot Hand Control System



By using our proposed SDK, we believe that we will be able to develop this robot hand systems efficiently and respond quickly to diverse customer demands.

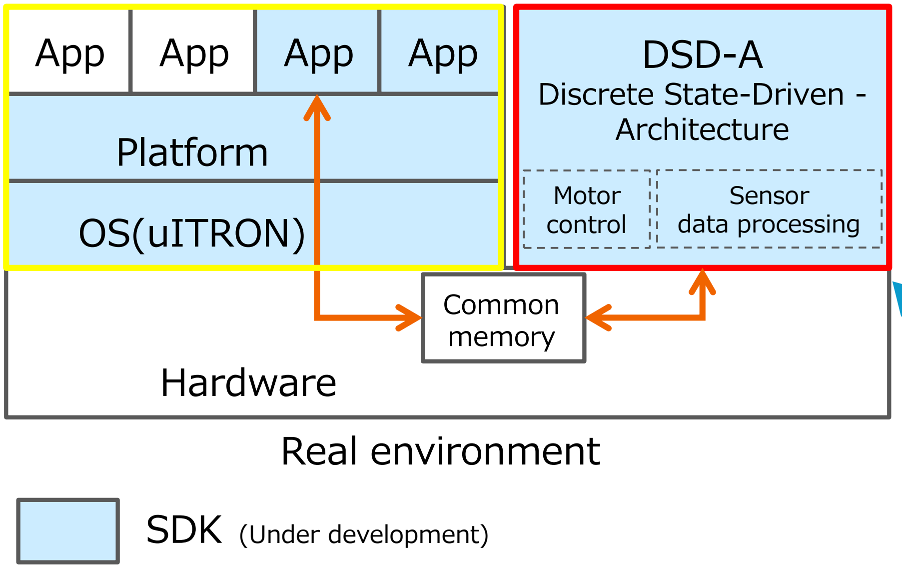
1.5 About Model-Based Design(MBD)



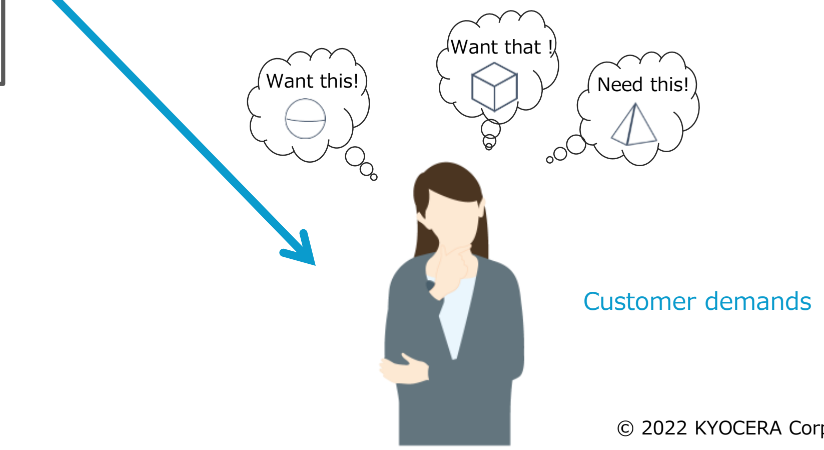
+ By using MBD, front-loading the designing process can be done which then improves the design quality.
+ Also, robustness can be estimated by considering variations and noise in virtual environment.
  Since there will be less “Rework” than before, development costs and resources can also be cut down.

2. Software Development Kit(SDK) for Robot Hand Control System

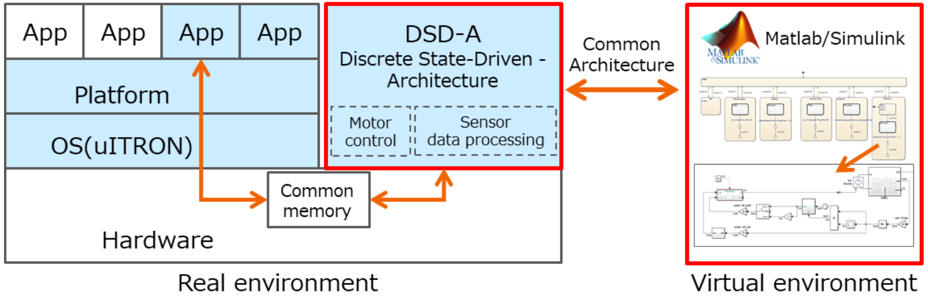
2.1 Features of SDK for Robot Hand Development



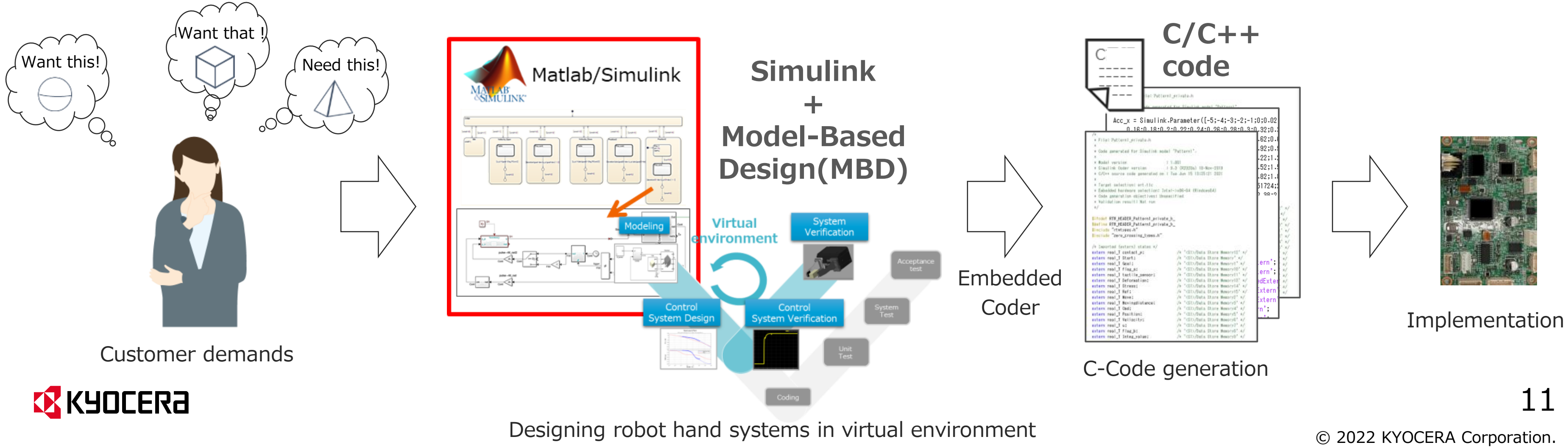
- The SDK is divided into DSD-A part, which performs sensor processing and controls actuators such as motors, and OS part, which manages other applications.
- The DSD-A and OS parts exchange information via common memory.
- DSD-A part handles the design of control software related to the hardware configuration, such as the type of sensors or motor control, used and modified according to customer needs.



2.2 Robot Hand Dev. Process using DSD-A in SDK Environment

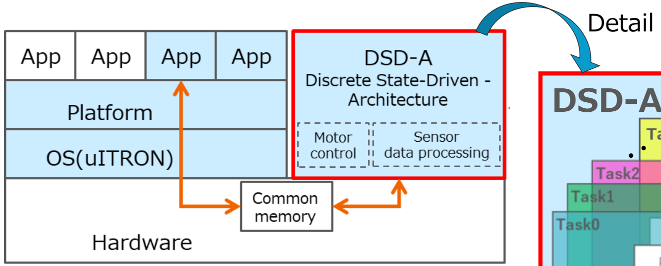


+ DSD-A part can be described by Simulink and has a common architecture in real and virtual environments.

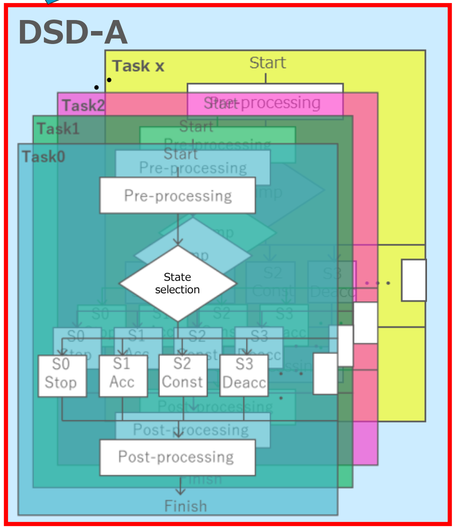


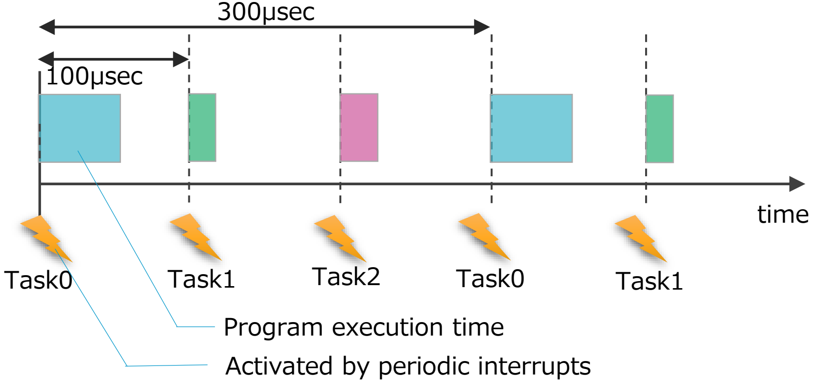
2.3 Features of Discrete State Driven-Architecture (DSD-A)

-Task Definitions and Discrete Periodic Processing



DSD-A part consists of multiple tasks

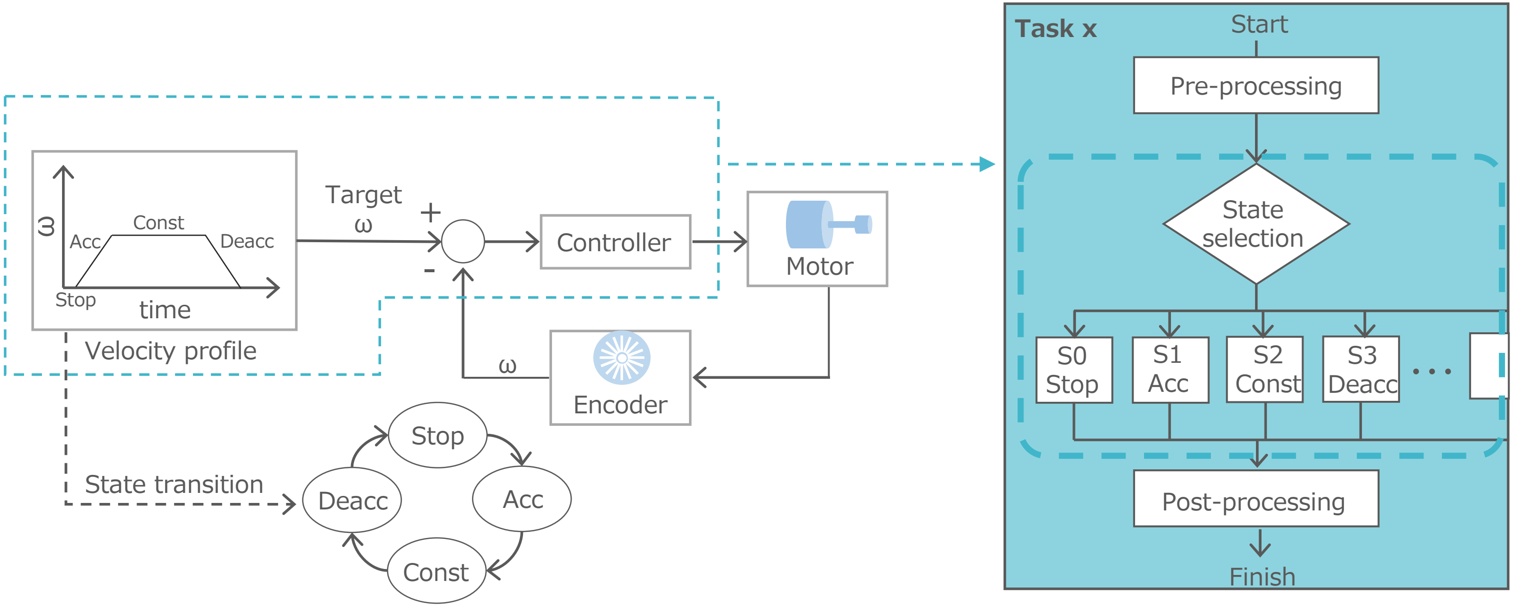




+ The sensors and actuators mounted on the robot hand are defined as task. For example, Actuator 1 is defined as Task 0, Sensor 1 as Task 1, Sensor 2 as Task 2, etc.
+ Each task is activated by periodic interrupts

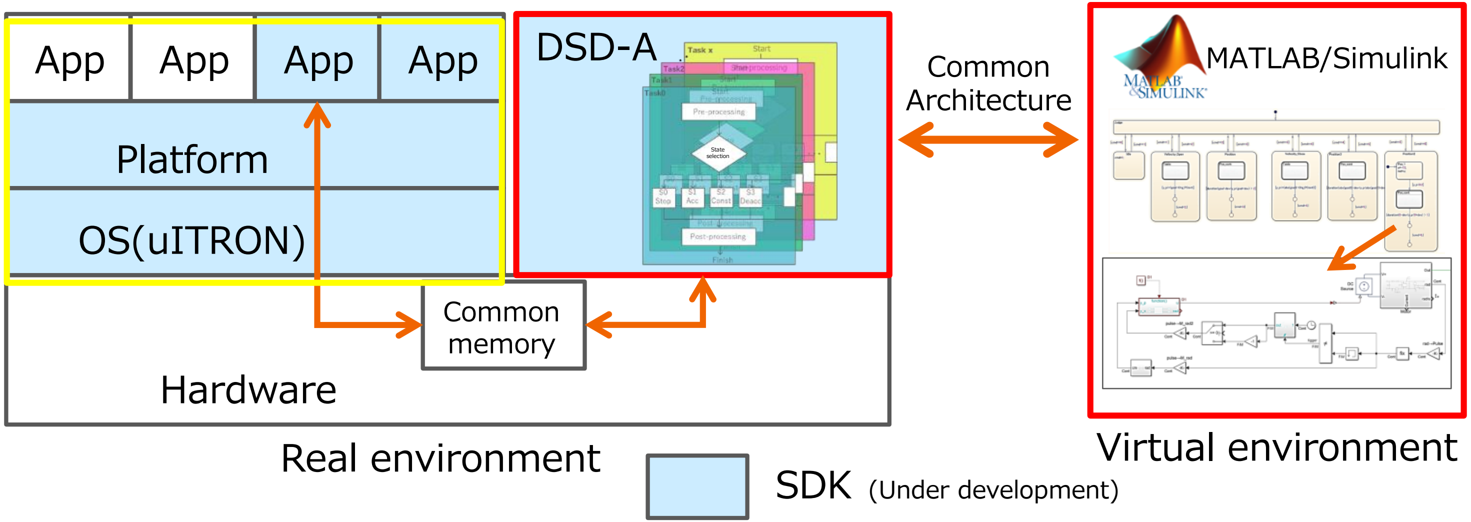
2.4 Features of Discrete State Driven-Architecture (DSD-A)

-State-Driven Control and Real-Time Assurance



+ The program is divided for each possible state of the actuators and sensor. Also, as the status transition from one to another periodically, the processing time of the program can be controlled, thus real-time processing can be assured.

2.5 Summary of SDK and DSD-A for Robot Hand Development



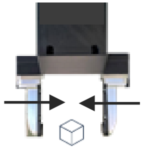
+ By using our proposed SDK, we believe that we will be able to develop this robot hand systems efficiently.
  Also, since DSD-A has common architecture in both real and virtual environments, it is possible to seamlessly implement the content designed in the virtual environment on to microcomputer.
+ With DSD-A, real-time processing can be assured.

3. Overview of Robot-Hand Development using Model-Based Design (MBD) and MATLAB® Tools

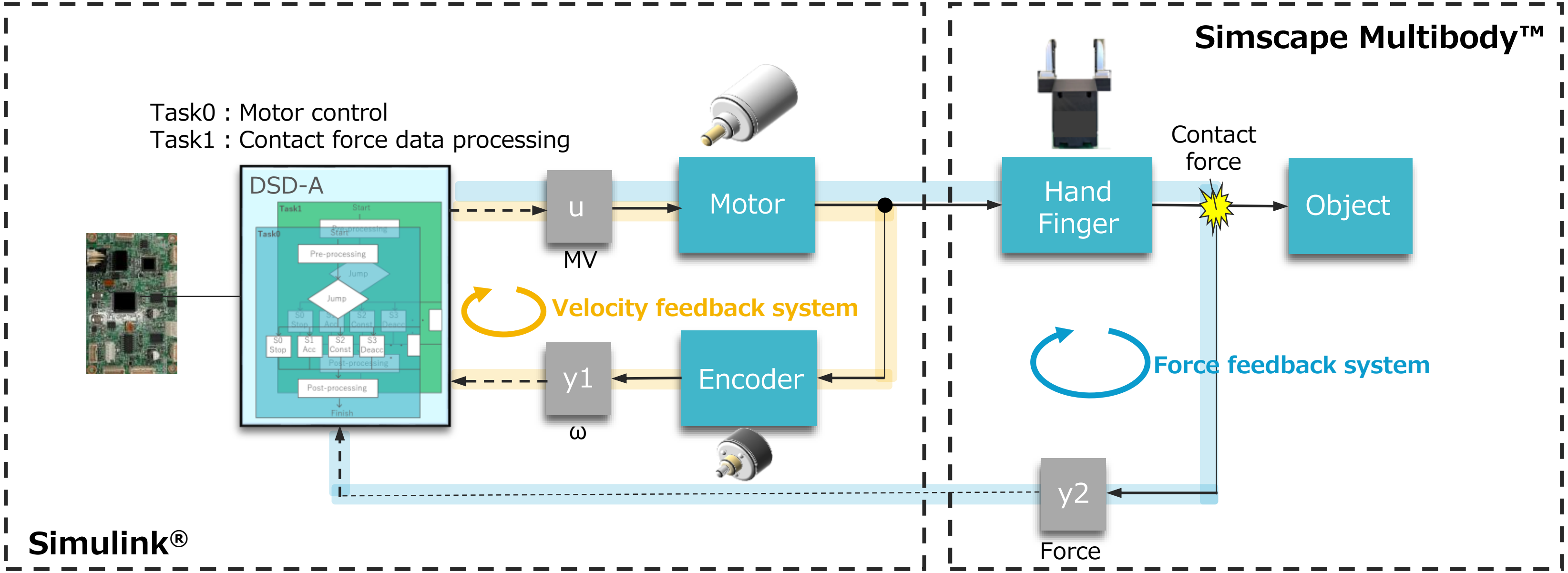
3.1 Robot Hand System Block Configuration

Robot Hand System

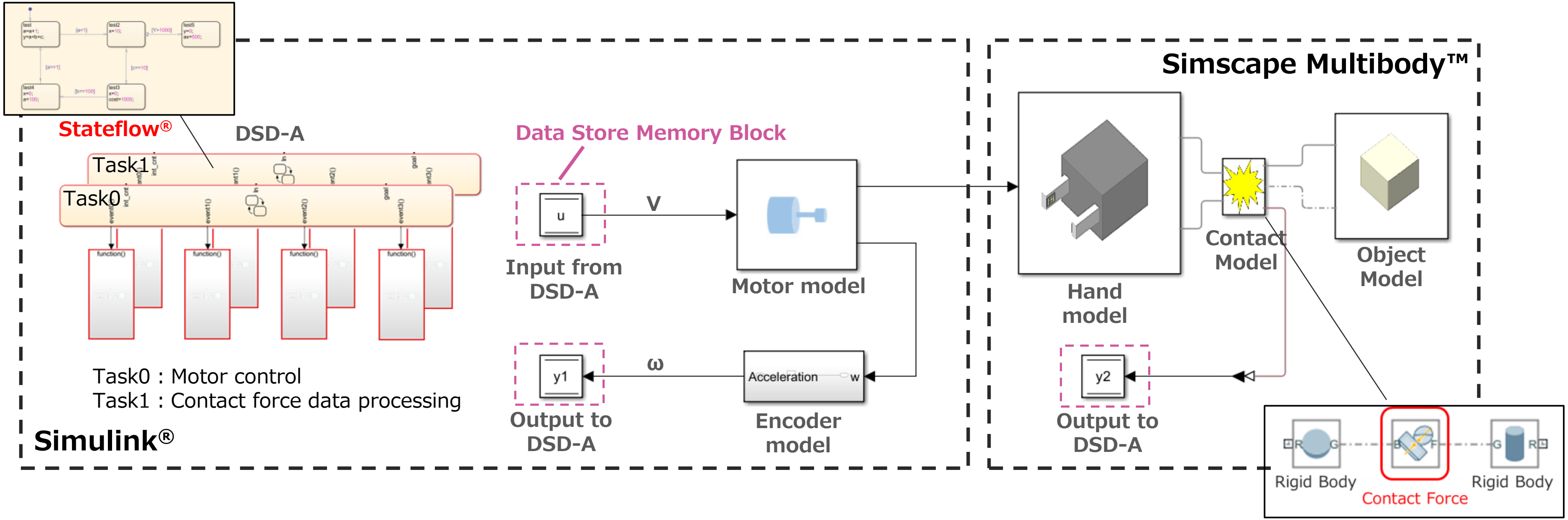
Robot hand system that enables stroke-width control of parallel-chuck robot finger by acquiring real-time information of velocity and contact force.



Parallel-chuck hand



3.2 Plant Modeling



+ The robot hand components drawn in the block diagram are modeled by Simulink® and Simscape™ Multibody™.
  For example, a contact force model can be expressed using a contact force block.
  DSD-A is described by Stateflow®, and task scheduling and control system design are performed.

3.3 DSD-A Design -Target Setting-

Target setting of control cycle and processing time

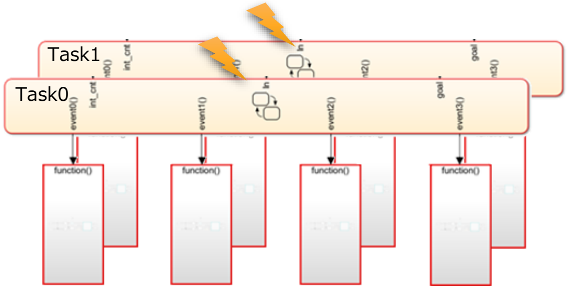
DSD-A Design Items:

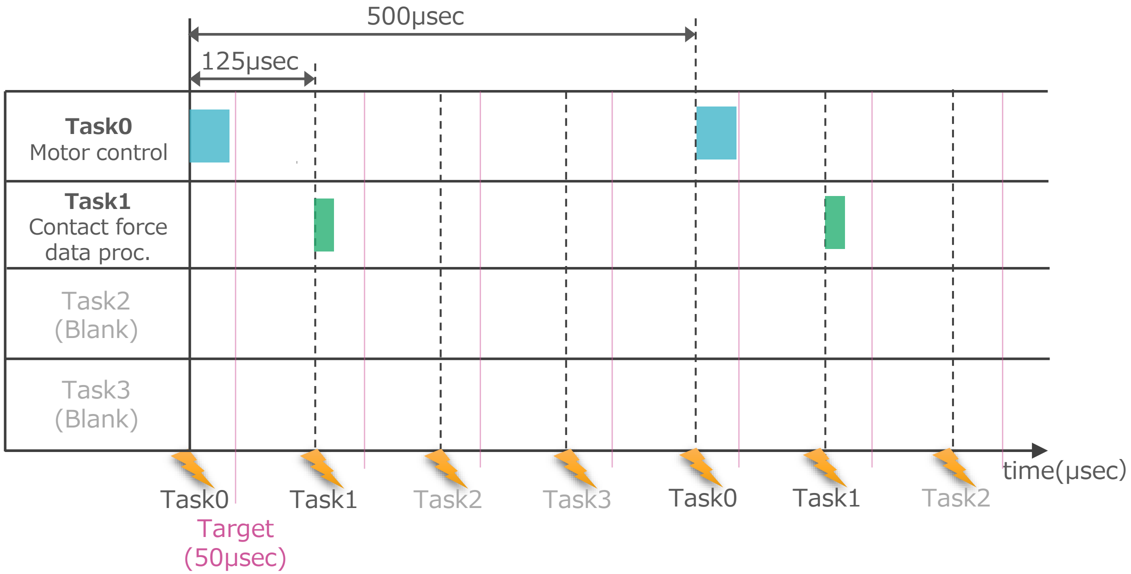
(Example)

・Control cycle : 500μsec

・Target processing time : 50μsec

・Processing time interval : 125μsec



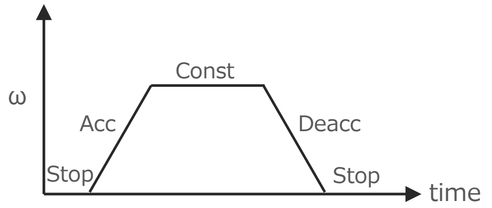


(: Execution timing)

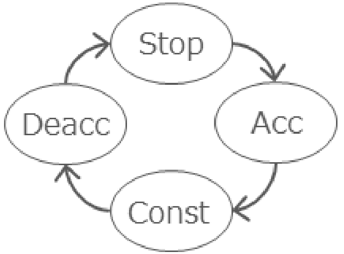
3.4 DSD-A Design –Assigning States-

Assigning states to motor control

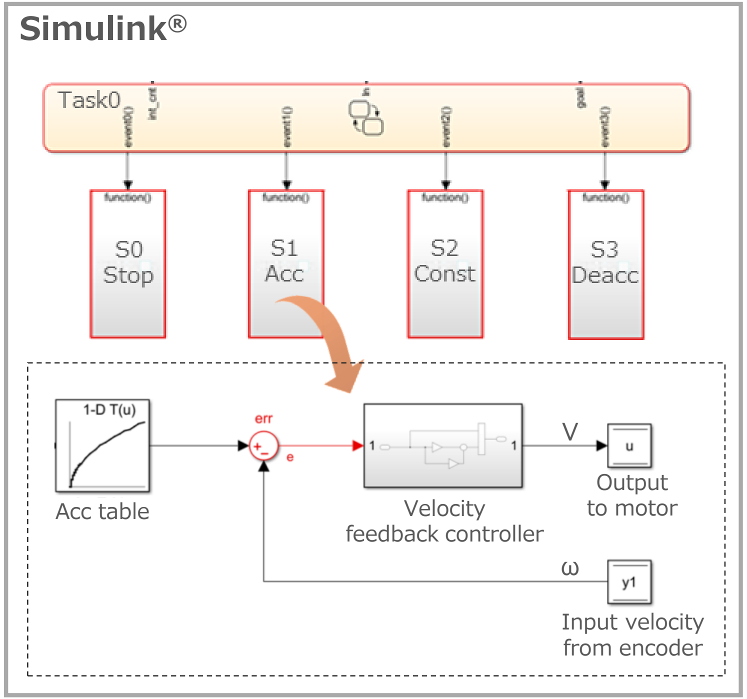
Ex.
Case of velocity feedback control



Velocity profile of motor



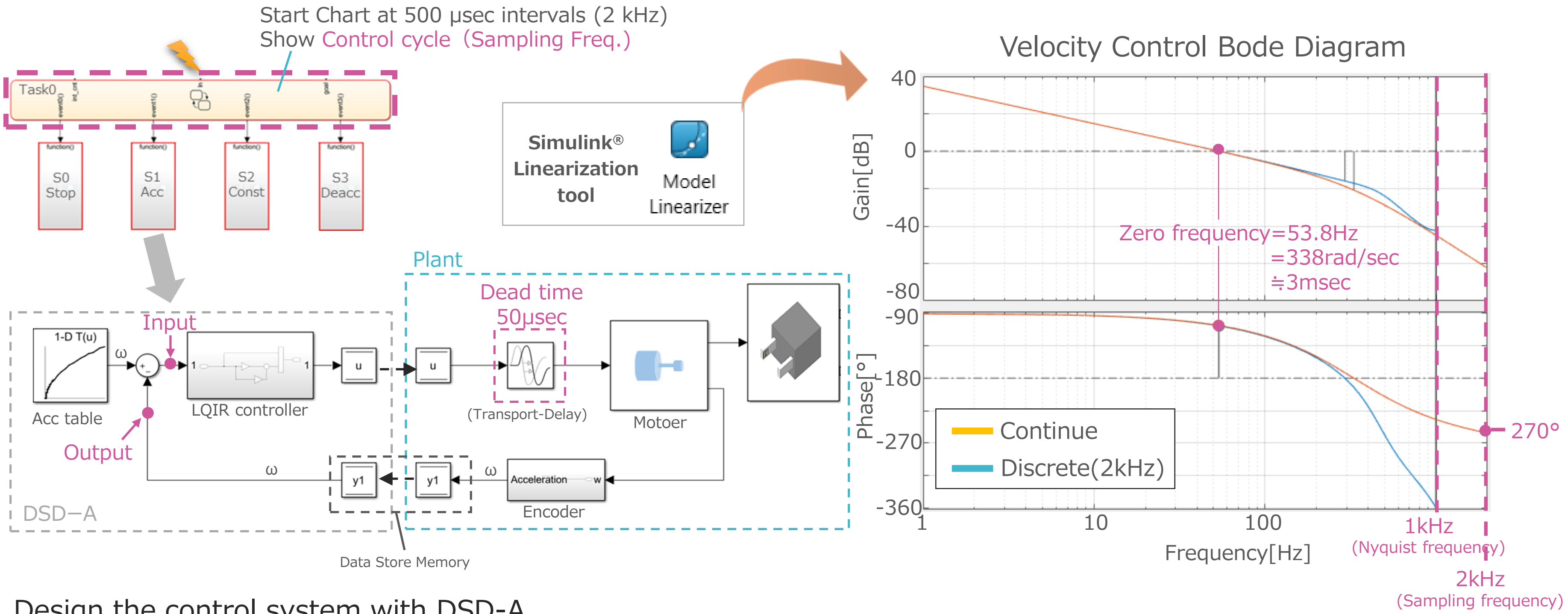
Ref: Motor velocity control state transition



Assign states to DSD-A based on state transitions

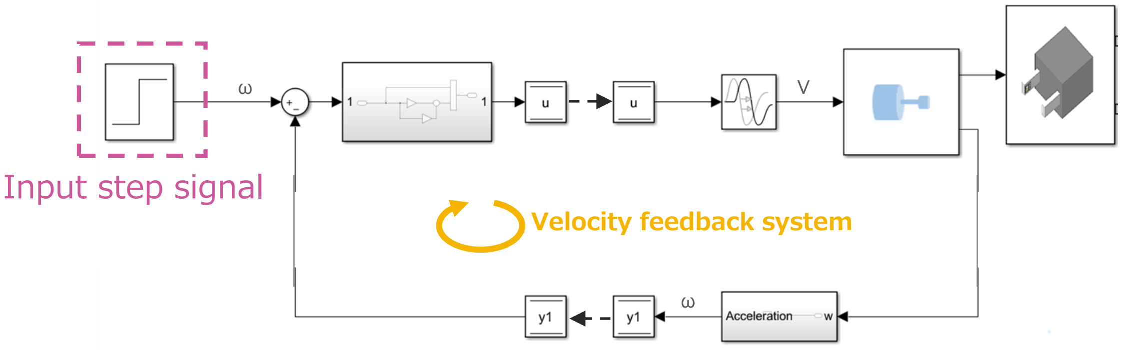
Assign Task 1 (Contact Force Data Processing)

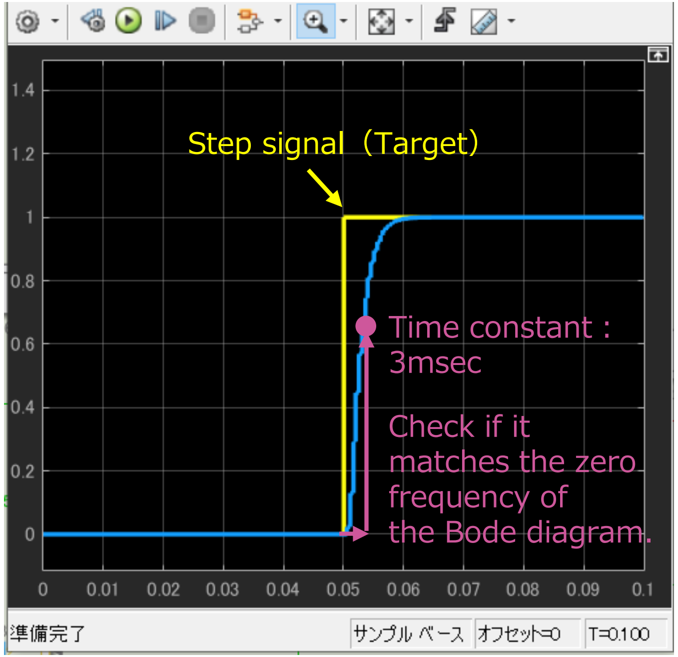
3.5 DSD-A Design -Velocity feedback controller design-



+ Design the control system with DSD-A
+ The control system can be designed to include sampling frequency and dead time
+ Use Simulink Linear Analysis tools to easily generate Bode diagram

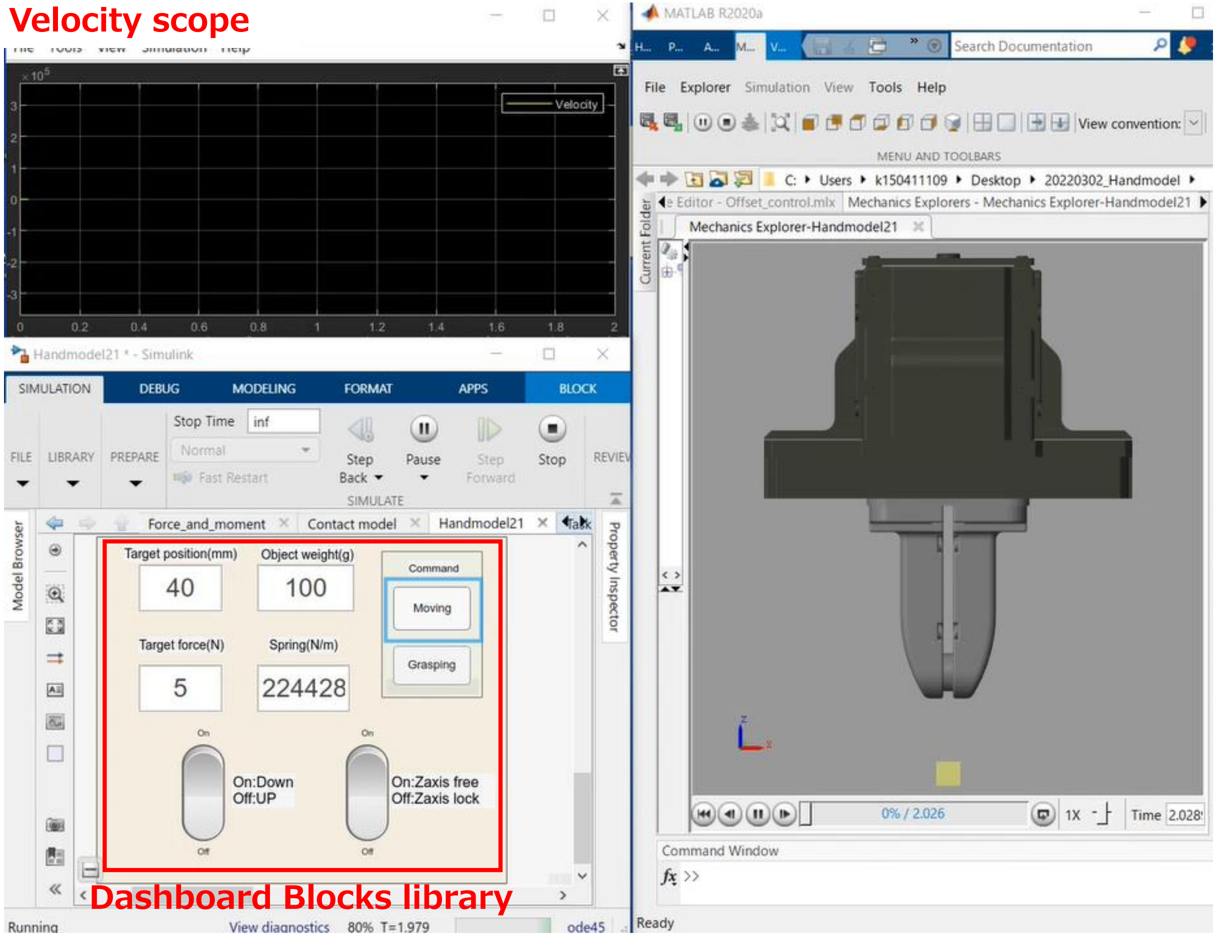
3.6 Verification - Velocity control system





Verification of speed control step response

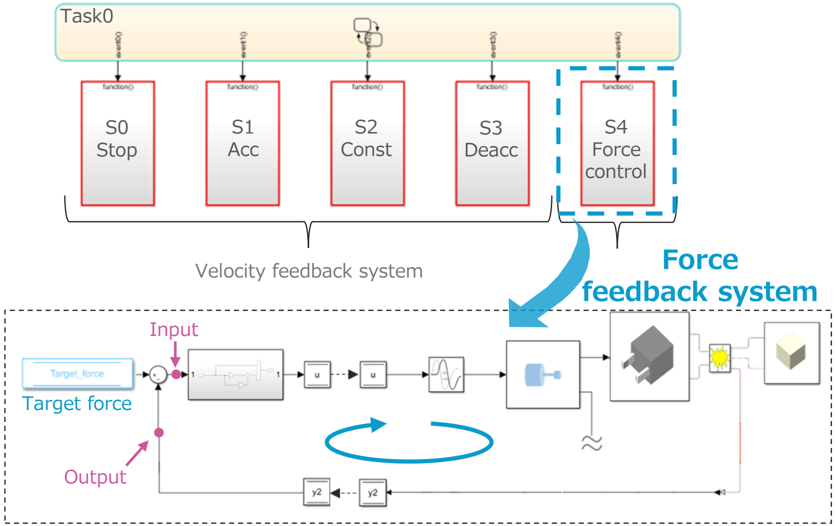
3.7 Visualization of System Dynamics -Velocity control system-

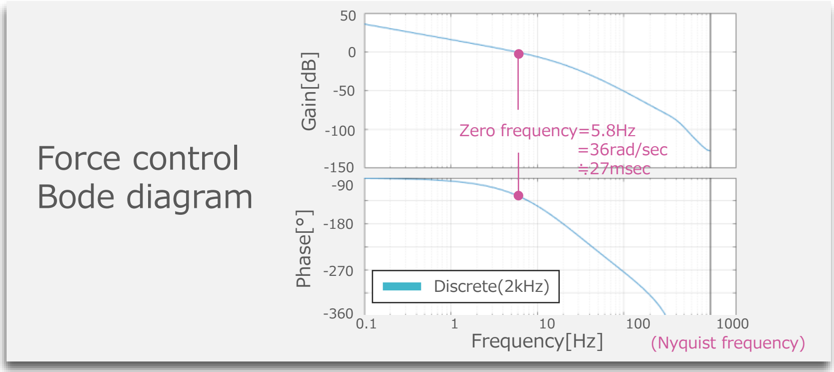


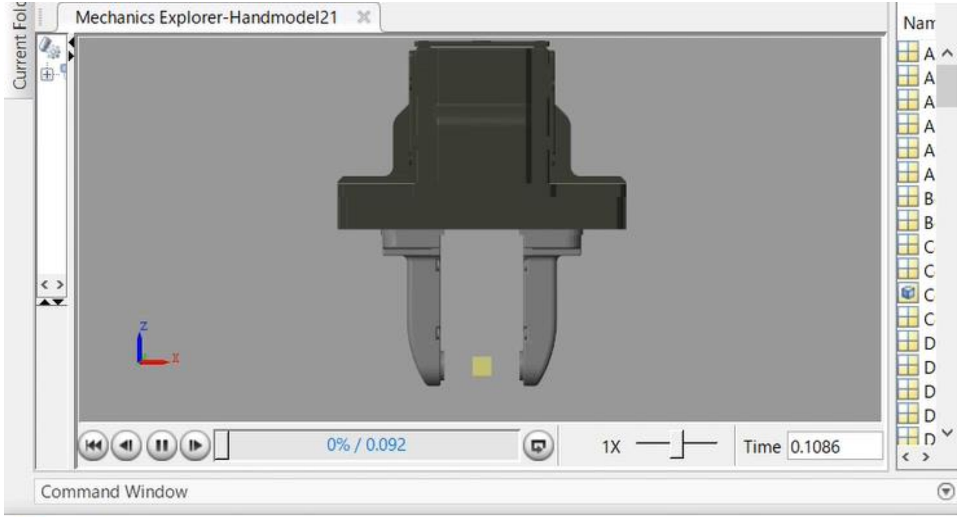
+ Simscape Multibody enables visualization of system dynamics

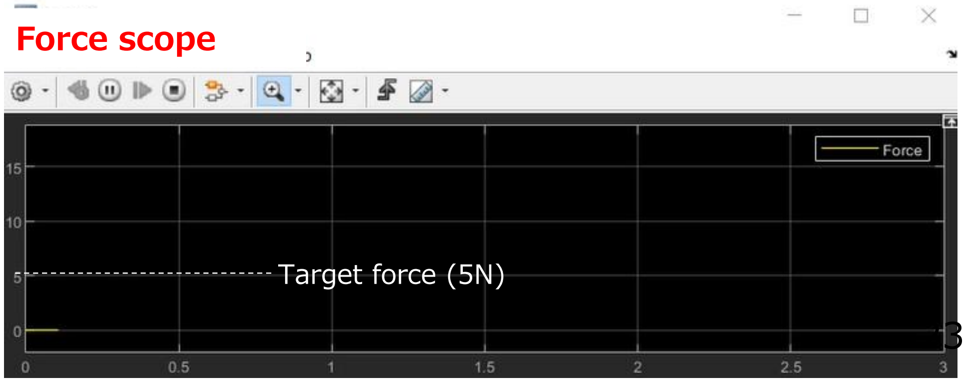
+ Dashboard Blocks library enables model manipulation and modification during simulation

3.8 Visualization of System Dynamics -Force control system-



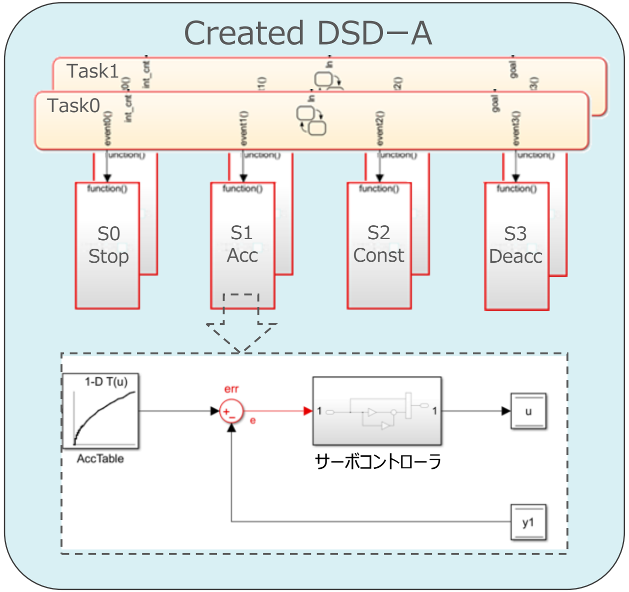




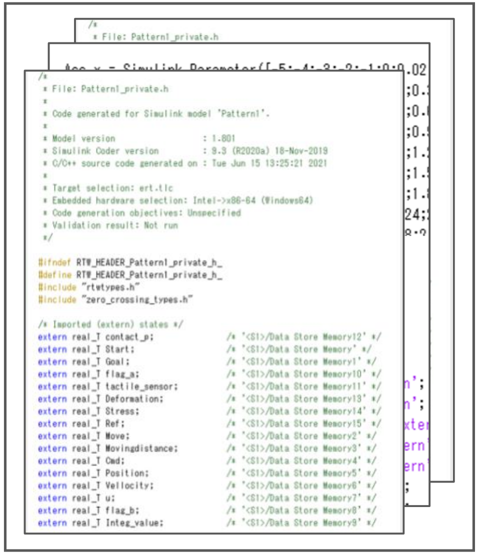


3.9 Code Generation using Embedded Coder and Implementation to MCU

Created DSD−A



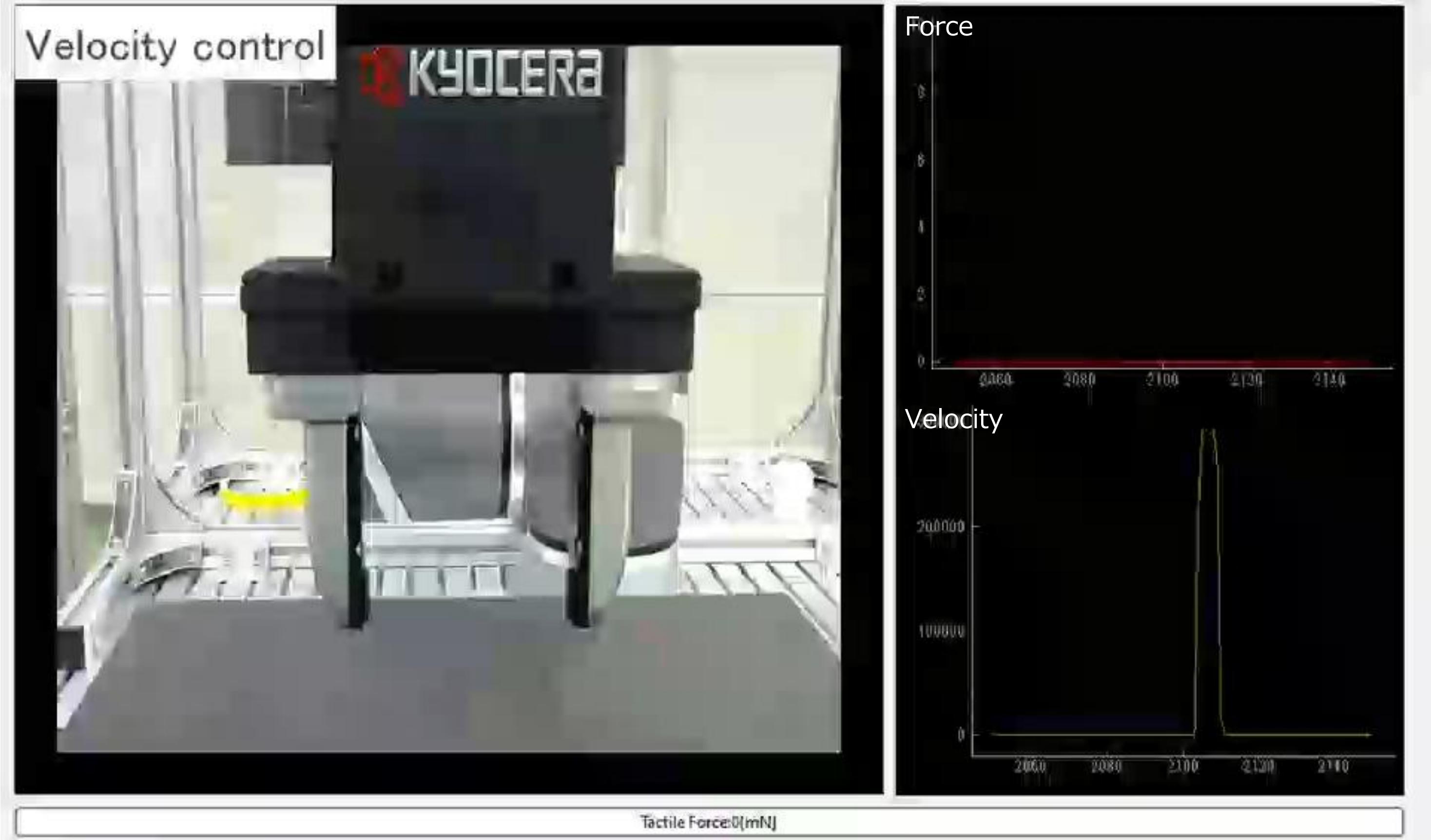
Code Generation



Generated code using MATLAB®
Embedded Coder

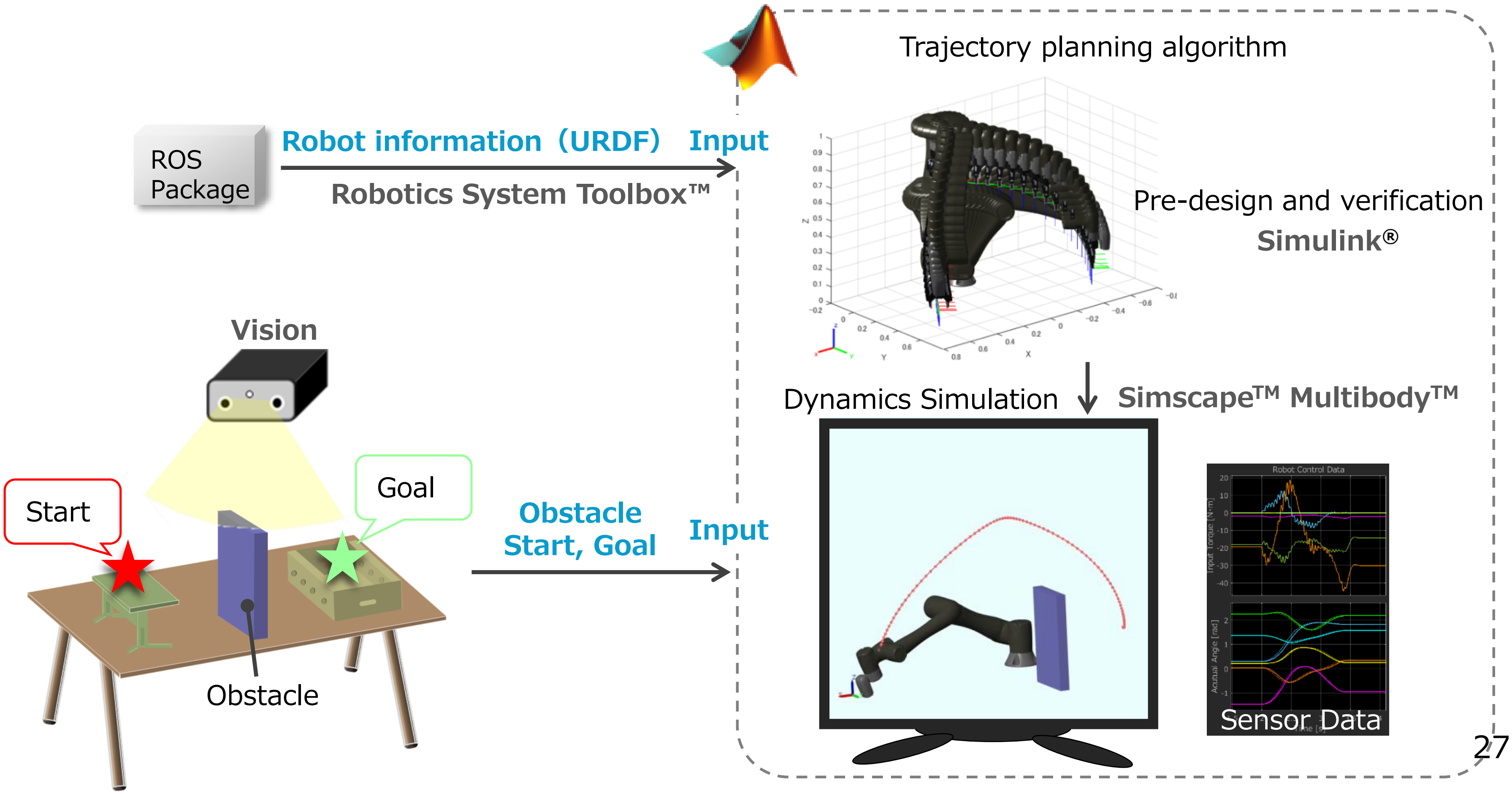
Implement

3.10 Validation on Hardware

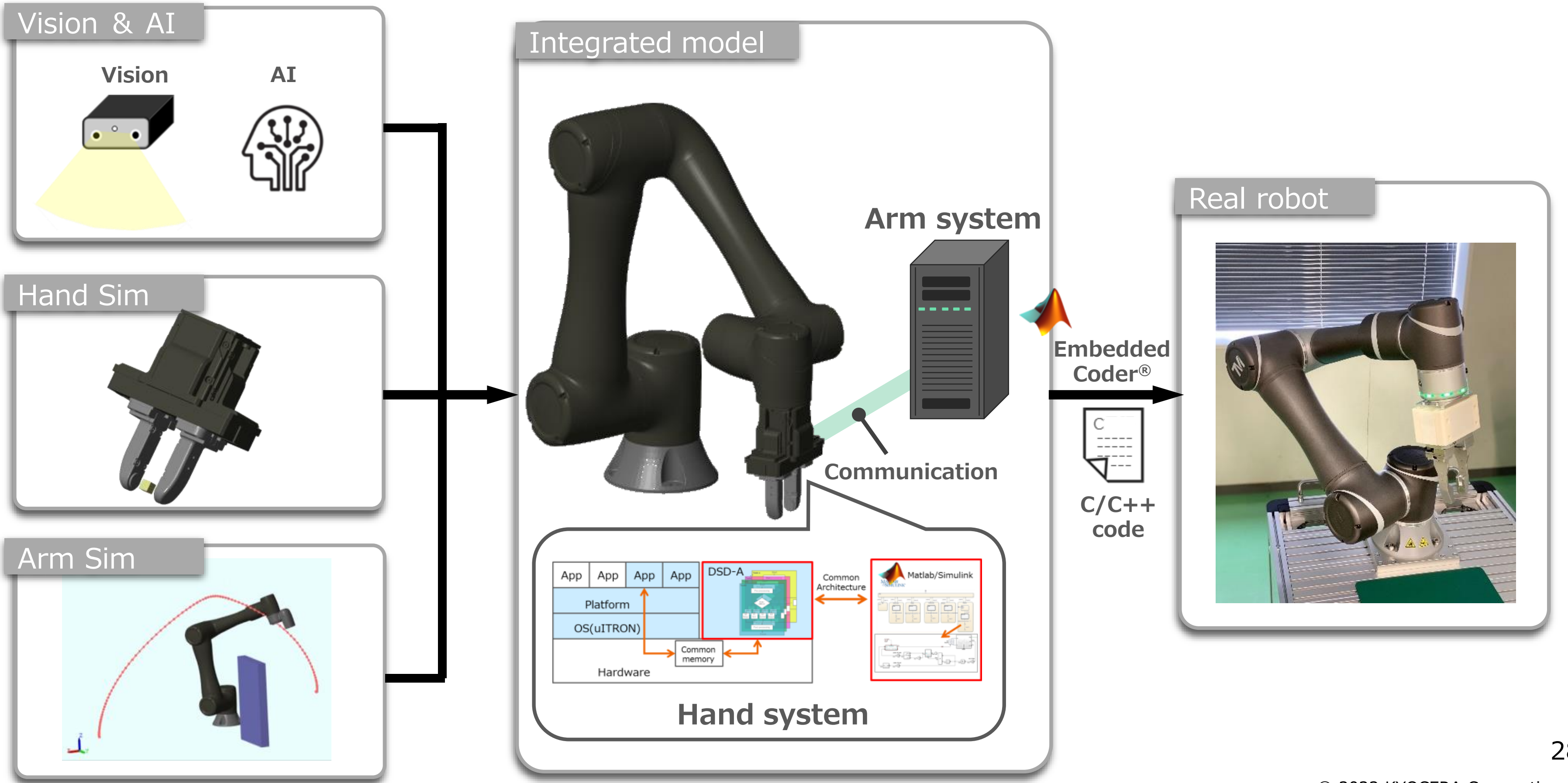


4. Construction of Controllable Multimodal Robot System

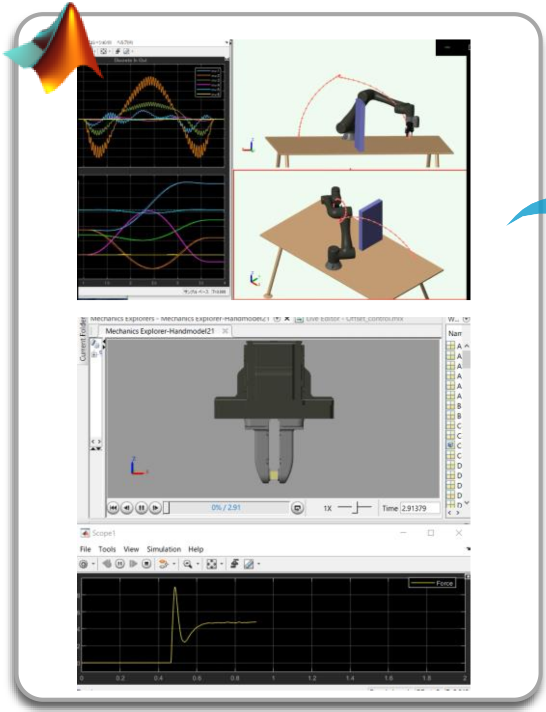
4.1 Overview of Robot-Arm system development



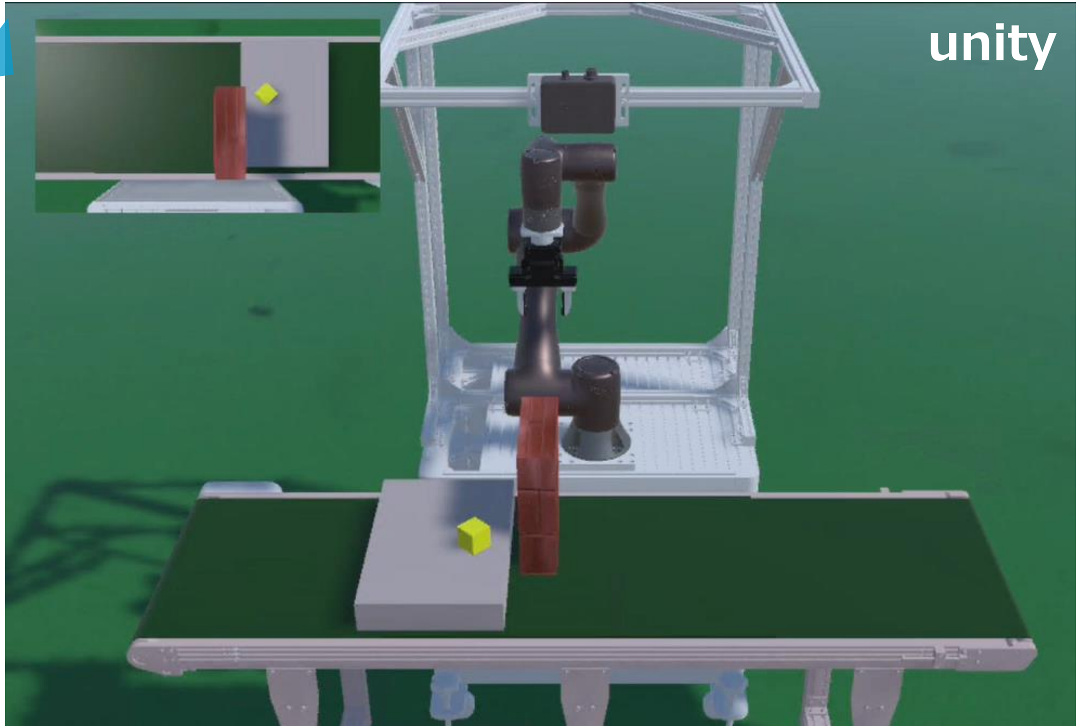
4.2 Integrating Robotic Arm and Hand Model -Construction of Multimodal System-



4.3 Integrating Robotic Arm and Hand Model -Construction of Multimodal System-



Simulation
Calculation result



5. Impression and Conclusion

5.1 Impression -MATLAB® used in Robotics-

■ Usability

+ Simulink® UI contains components that enable users (even beginners) to create and build their own system easily
+ Plenty of apps such as *Linearization Manager* and *Model Linearizer* to help users design their control systems
+ Simscape™ Multibody™ visualizes the robot system dynamics and help users troubleshoot control system errors
+ Embedded Coder® allows users to generate C and C++ code automatically to reduce costs and resources in developing software

■ Request

+ Improving performance for complex simulations (e.g. contact forces, multiple sensors, robot complexity, etc.)

  ⇨ I hope MATLAB® will provide tools to understand tradeoff between performance and precision

5.2 Conclusion

+ We are developing an autonomous collaborative robot capable of multi-modal control based on camera, arm, hand, and various sensor information in order to meet diverse customer needs

+ This session focused on an overview of our development of robot hand with multiple sensors based on MBD. We also introduced SDK and DSD-A design to improve the efficiency in robot hand development.

+ We believe that we can increase the development speed of robot hands by using the proposed development process that combines the SDK and DSD-A for robot hands with the MBD and MATLAB® roots.
  Hopefully, this problem could lower the difficulty for development in robotics industry.

END