Benchmark Problem for Motion Control of Four in-Wheel Motor Actuated Vehicles

Yutaka Hirano, Rui Gao, Junichi Kako, Fuguo Xu, and Tielong Shen

***Abstract*— This article proposes a benchmark problem for the challengers about optimization of body motion control and energy consumption of Electric Vehicles (EVs) equipped with 4 In-Wheel Motor (IWM). By using IWM, not only the control of driving and braking behavior but also the control of the body attitude of EV become possible. In this paper, the challenging problem that includes two subproblems is described firstly. For the challengers, a full 4-IWM EV simulator written by Modelica is provided. A Functional Mockup Unit (FMU) of the full vehicle model for connecting between Modelica and MATLAB/Simulink is also provided. Challengers are requested to design their own controller in Modelica or Simulink.**

***Index Terms*— Vehicle dynamics control, powertrain control, In-Wheel Motor, electric vehicle, modelica, FMU, Simulink**

I. INTRODUCTION

Autonomous and electrification are two important trends in the field of automotive industry, and motion control for the electrified vehicles is fundamental technology to implement autonomous driving. However, in the academic field of control and decision, this issue has not been attracted much attention although this problem is tackled in the last few decades in the field of vehicular engineering.

The aim of this article is to propose a benchmark problem of motion control for the electrified vehicle. This benchmark problem provides a simple, yet meaningful, motion control problem formulation for four in-wheel motor (IWM) actuated electric vehicles (EV), which is motivated by the fundamental technology of autonomous driving. The team of proposers consists of experts of industrial researchers and academic field of control.

The benchmark problem is divided into two phases, so that the challengers can attack the problem step-by-step. A uniformed evaluation index will be given.

Challengers will be provided the following environments for challenging the problems; 1) A simulation model of targeted vehicle which is provided in the form of Modelica module with free license of half a year. 2) Simulation platform combined the vehicle model and control module in MATLAB/Simulink is also provided such that the challenger is able to implement the designed control law in Simulink module. (Simulink license will not be provided.)

The rest of this article is organized as follows. In Section II, outline and definition of the problems to be solved of this benchmark problem are proposed. Details of the full vehicle model by Modelica is introduced in Section III. Also, a simple sample controller by the authors in the Modelica environment as a reference is introduced in Section III. Simulink models including FMU (Functional Mockup Unit) by FMI (Functional Mockup Interface) standard exported from the Modelica model are introduced in Section IV. Also a default controller by the authors for the Simulink model is introduced in Section IV.

II. OUTLINE OF THE BENCHMARK PROBLEM

The benchmark problem is to control the vehicle dynamics for 4-wheel IWM vehicle. It has been shown that 4-IWM vehicle can control not only the driving/braking behavior but also can control the body attitude by controlling the vertical reaction force at each wheel caused by anti-dive geometry and anti-squat geometry of the suspension [1]. This benchmark problem is formulated to control both of the vehicle driving state and the vehicle body attitude with minimum energy consumption by controlling the torque of the four IWM of each wheel independently.

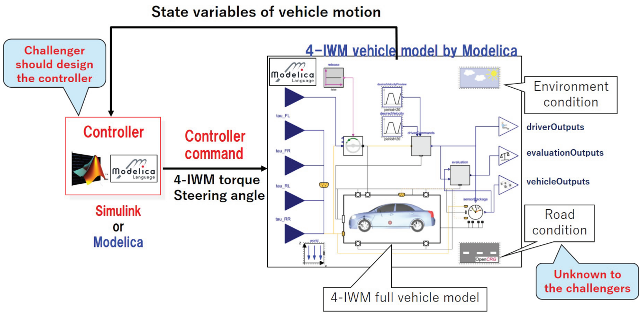


Fig. 1. The flamework of the benchmark problem.

Fig. 1 shows the flamework of this benchmark problem. The full vehicle model of the 4-IWM vehicle is provided as a Modelica model including 3D multibody mechanical model of the suspension and body, electric model of battery and motor, and non-linear tire model. The challengers are requested to design and test a controller of 4-IWM torque and steering angle. The controller should be implemented by Simulink or Modelica. There are two tasks to be solved in this benchmark problem.

This work was not supported by any organization

Y. Hirano is with HIRANO Research Labo, Tokyo 101-0021, Japan `yutaka.hirano.wild7@gmail.com`

R. Gao is with Modelon K.K., Tokyo 101-0041, Japan `rui.gao@modelon.com`

J. Kako is with Higashi-Fuji Technical Center, Toyota Motor Corporation, Shizuoka 410-1107, Japan `junichi_kako@mail.toyota.co.jp`

F. Xu and T. Shen are with Department of Engineering and Applied Sciences, Sophia University, Tokyo 102-8554, Japan `fuguoxu@ieee.org,tetu-sin@sophia.ac.jp`

Problem 1 (Acceleration and braking on rough slippery straight road): Minimizing deviation from the desired speed profile, body motion (heave, pitch, roll) and energy consumption.

Problem 2 (ISO double lane change on rough road): Minimizing deviation from the desired course, body motion (heave, pitch, roll) and energy consumption.

The undulation of the rough road is unknown to the challengers. Final evaluation of the benchmark problem will be done using the different road undulation pattern provided by the organizer.

A. *Definition of Problem 1*

Task of acceleration and braking on rough slippery straight road is given as the Problem 1. It is requested that the vehicle follows the desired speed profile as shown in Figure 2. Friction coefficient of the road surface is 0.5.

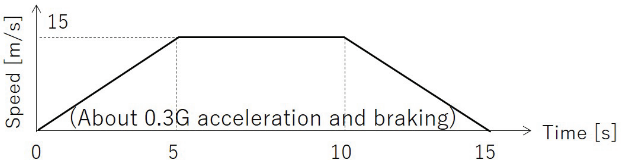


Fig. 2. Desired vehicle speed profile for the Problem 1.

Evaluation indexes for the Problem 1 are settled as bellows.

Requirement 1-1: Reduce the vertical body motion to below the targets.

$$|a_z(t)| \leq a_{z_{Target1}} \tag{1}$$

$$|\theta(t)| \leq \theta_{Target1} \tag{2}$$

$$|\psi(t)| \leq \psi_{Target1} \tag{3}$$

where a_z, θ and ψ are the vertical acceleration, roll angle and pitch angle, respectively.

Requirement 1-2: Minimize the deviation of the vehicle speed from the target.

$$J_{v1} = \int_0^{t_e} \left\{ v(t) - v_{Target}(t) \right\}^2 dt \tag{4}$$

where t_e is th total running time, v is the vehicle speed.

Requirement 1-2: Minimize the energy consumption.

$$J_{e1} = \int_0^{t_e} E(t)dt \tag{5}$$

$$E(t) = \sum_{i=1}^{4} V_B(t) \cdot I_i(t) \tag{6}$$

where E, V_B and I_i denote the power consumption, battery voltage and current of each IWM.

B. *Definition of Problem 2*

Task of ISO double lane change on rough road is given as the Problem 2. It is requested that the vehicle follows the desired path of ISO double lane change course at vehicle speed of 60[km/h]. Friction coefficient of the road surface is 0.9.

Evaluation indexes for the Problem 2 are settled as bellows.

Requirement 2-1: Reduce the vertical body motion to below the targets.

$$|a_z(t)| \leq a_{z_{Target2}} \tag{7}$$

$$|\theta(t)| \leq \theta_{Target2} \tag{8}$$

$$|\psi(t)| \leq \psi_{Target2} \tag{9}$$

Requirement 2-2: Minimize the deviation of the vehicle trajectory from the target.

$$J_{v2} = \int_0^{t_e} \left[\left\{ x(t) - x_{Target}(t) \right\}^2 + \left\{ y(t) - y_{Target}(t) \right\}^2 \right] dt \tag{10}$$

where x and y denote the vehicle longitudinal position and vehicle lateral position, respectively.

Requirement 2-3: Minimize the energy consumption.

$$J_{e1} = \int_0^{t_e} E(t)dt \tag{11}$$

$$E(t) = \sum_{i=1}^{4} V_B(t) \cdot I_i(t) \tag{12}$$

III. Provided Simulation Platform

A. *Vehicle Model by Modelica*

Vehicle dynamics model is provided as a Modelica model [2]. As a Modelica tool, Modelon Impact is provided with free license for the challengers until the end of 2023[3].

Fig. 3 shows the 1st layer and the 2nd layer of the Modelica full vehicle model. The vehicle model consists of chassis model by 3-dimentional multibody mechanical system model and power system model of IWM drive train by electric model. Brake model by hydraulic circuit model is not used. The driverCommand model calculates driver's maneuver about steering, acceleration and break. Some outputs of the driverCommand model are just used as a reference of the target signals of the vehicle motion for a default controller.

Fig. 4 shows the construction of the chassis model and the power system model. (Each model is the 3rd layer of the full vehicle model.) Chassis model consists of the models of front chassis suspension kinematics, rear chassis suspension kinematics, Body motion and tires with non-linear characteristics. Tire model is based on Pacejka2002 implementation of the magic formula model[4]. The chassis model was developed based on Vehicle Dynamics Library (VDL) of Modelon Impact. On the other hand, the Power system model consists of battery system model, 4 IWM

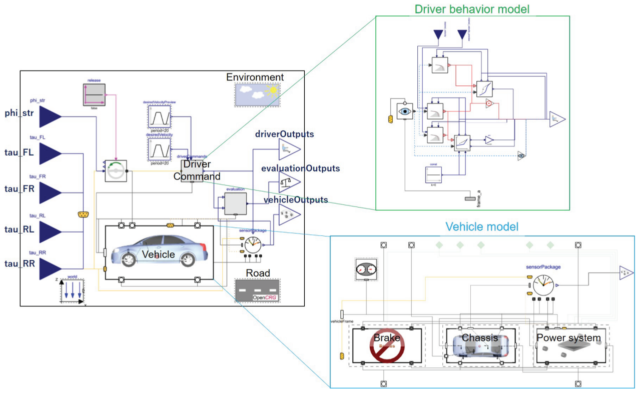


Fig. 3. 1st and 2nd layer of Modelica full vehicle model.

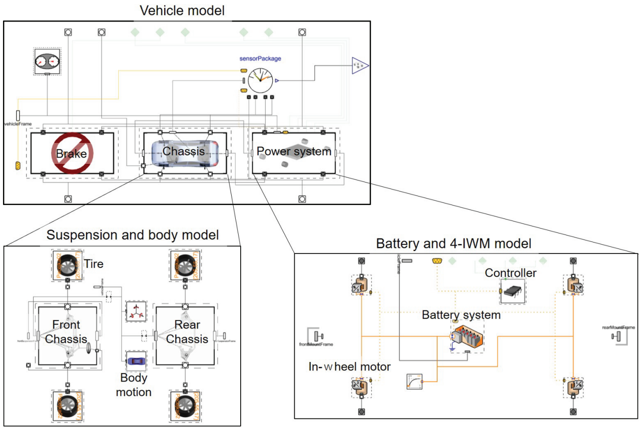


Fig. 4. Construction of the chassis model and the power system model.

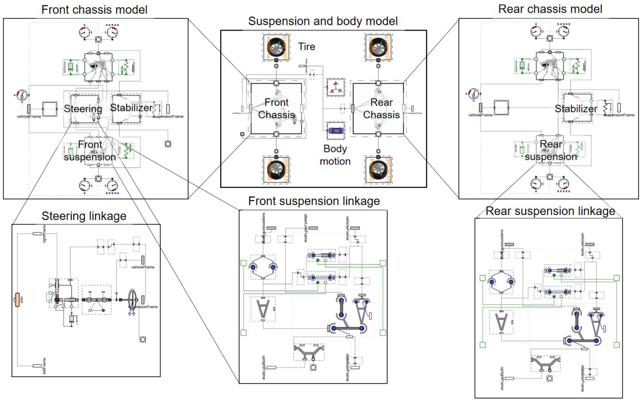


Fig. 5. Details of the chassis suspension and body model.

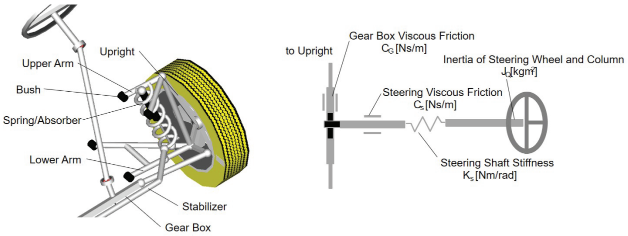


Fig. 6. Details of the front suspension linkage and steering linkage model.

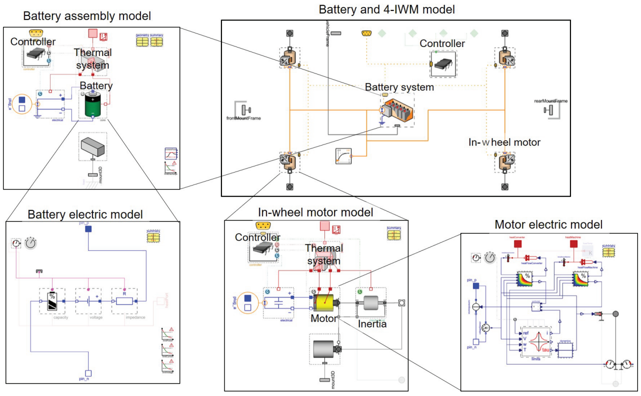


Fig. 7. Details of the battery and 4-IWM model of the power system.

models and the controller model. It was constructed by utilizing Electrification Library of Modelon Impact.

Fig. 5 shows the construction of details of the chassis suspension and body model. Front suspension model consists of front suspension linkage, steering linkage and stabilizer linkage. Details of the front suspension linkage and steering linkage model is displayed in Fig. 6. Both front suspension and rear suspension are conventional double wishbone suspension. Each suspension model is provided with its coordinate values of every linkage hard points. Also, non-linear characteristics of suspension bushes are given. In the steering linkage model, the effects of stiffness and viscous friction of the steering shaft and steering lack are also considered. Rear chassis model is almost same as the front chassis model, but does not have the steering linkage. Please note that the value of actual steering angle and steering angle command from the controller are strictly not identical because of steering shaft stiffness and friction. These command error should also be considered when designing the controller by challengers.

Fig. 7 is details of the power system model with battery system and 4 In-wheel motor models. Battery system model consists of the battery electric model, battery controller model and also a model of thermal behavior caused by electric losses. In-wheel motor model consists of a motor electric model and mechanical inertia model also with the thermal behavior model.

By constructing the hierarchical model of the full vehicle model as written above, Modelica translator tool (this time, Modelon Impact) solves the simultaneous differential algebraic equations (DAEs) and generate the simulation codes automatically. Finally, the simulation results of each physical variable can be calculated and plotted. Also, by using Modelica tool, the animation results such as shown in Fig. 8 can be generated and displayed.

As shown in Fig. 3, there are connectors for input and output to the vehicle model in the 1st layer of the model. As the connectors for input, scalar signals shown in Table I are given from outside of the model, for example, from a controller model. As the connectors for output, signal vector defined as ‘driverOutputs’, ‘evaluationOutputs’ and ‘vehicleOutputs’ are connected. Each signal vector has members of the signals

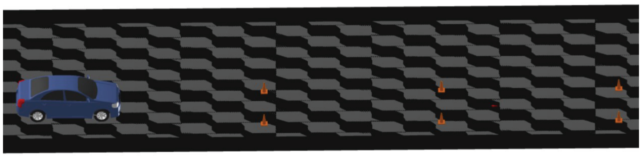


Fig. 8. Example of animation result of simulation by Modelica (Double Lane change).

TABLE I

DEFINITION OF INPUT CONNECTOR SIGNALS

Meaning	Signal name [Units]
Steering wheel angle command	ϕ_{str} [rad]
FL wheel torque command	τ_{FL} [Nm]
FR wheel torque command	τ_{FR} [Nm]
RL wheel torque command	τ_{RL} [Nm]
RR wheel torque command	τ_{RR} [Nm]

defined in Table II, Table III, and Table IV respectively. Here, as abbreviations of tire position, following symbols are used as FL(fl): Front Left, FR(fr): Front Right, RL(rl): Rear Left and RR(rr): Rear Right.

The 'driverOutputs' connector includes signals from the driver behavior model of VDL as defined in Table II.

The 'evaluationOutputs' connector includes signals about the evaluation criteria of Problem 1 and Problem 2 as defined in Table III. These signals are used to evaluate the performance of the controller designed by each challenger.

The 'vehicleOutputs' connector includes signals about the vehicle motion status as shown in Table IV. These signals can be used for the controller design and monitoring.

TABLE II

DEFINITION OF 'DRIVEROUTPUT' CONNECTOR SIGNALS

Meaning	Signal name [Units]
CloseLoopDriver pathPoint reference velocity along path at t[k] and t[k-1]	$vV_{P_x}[2]$ [m/2]
CloseLoopDriver pathPoint x position	rV_x [m]
CloseLoopDriver pathPoint y position	rV_y [m]

TABLE III

DEFINITION OF 'EVALUATIONOUTPUTS' CONNECTOR SIGNALS

Meaning	Signal name [Units]
Evaluation index 1: $J_e = \int_0^{t_e} \{E(t)\} dt$ $E(t) = \sum_{i=1}^{4} V_B(t) \bullet I_i(t)$	J_e
Evaluation index 2: $J_{v1} = \int_0^{t_e} \{v(t) - v_{Target}(t)\}^2 dt$	J_{v_1}
Evaluation index 3: $J_{v2} = \int_0^{t_e} \left[\{x(t) - x_{Target}(t)\}^2 + \{y(t) - y_{Target}(t)\}^2\right] dt$	J_{v_2}

B. Exporting models for integration with Simulink

Modelica models in the *CDC2023.Vehicles.Exports* package can be exported as FMU (Functional Mockup Unit) to be imported and connected in Simulink environment. FMU is an executable unit by world-wide model interface standard FMI (Functional Mockup Interface). The file extension of FMU is .fmu and its content is same as ordinary zip file. FMUs are provided to challengers for Acceleration and DoubleLaneChange models for this benchmark problem. About the actual way of importing FMU and connecting it with Simulink model, please refer to part C in the Appendix.

IV. EXAMPLE CONTROLLER

A. Simulink models

The organizers provide challengers Simulink models for simulating above mentioned Problem 1 and Problem 2 including corresponding FMUs of the vehicle model developed by Modelica environment. Fig. 9 shows the whole diagram of the provided model. There is a subsystem including the FMU of the vehicle model. The part displayed by the red box is a default controller made by organizers. The default controller only aims to track the desired vehicle speed and the desired path by the control logic mentioned in Eq. 13 and Eq. 14. The challengers are requested to make their own controller to fulfil all the requirements mentioned in the Section II.

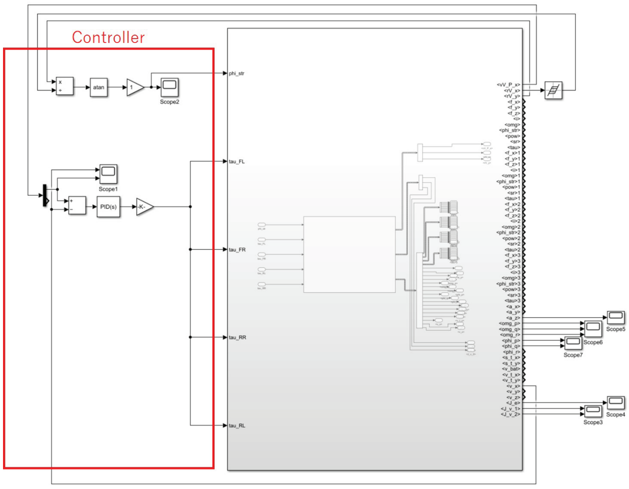


Fig. 9. Simulink model for the simulation.

$$\phi_{str} = \tan^{-1} \frac{rV_y}{rV_x} \times Gain \tag{13}$$

$$\tau_{**} = \left(K_p + K - I\frac{1}{s} + K_d \frac{N}{1 + N\frac{1}{s}} \right) (vV_{P_x}[1] - v_x) \tag{14}$$

where $**$ includes FL, FR, RL and RR.

B. How to use the Simulink model

To use the provided Simulink models, it is necessary to execute the following procedures every time the challengers will open the Simulink models and run the models at the first time in that session.

(Step 1) It is necessary to extract the information from the FMU and build an executable module (dll) from the source codes of the model involved in the FMU. To do this, double-click the FMU block in the Simulink model or just try to run the simulation of the Simulink model. (In the latter case, simulation fails because of lack of the procedure in the Step 2, but the dll will be made by this procedure.) Usually, it takes several minutes to finish this step. (Do not forget to add Matlab Path to the working directory.)

(Step 2) It is also necessary to validate the bus objects of the connectors of the FMU. To do this, just execute the following command in Matlab window.

fmudialog.createBusType(′ <Name of the Simulink model including subsystem name> / <Name of the FMU block>′)

In the case of the provided model for the Problem 1, the MATLAB command is as below and there will be responses as belows.

fmudialog.createBusType('CDC_Task1_Sample/Subsystem/FMU

Bus Object for Output Port 1: driverOutputs

Bus Object for Output Port 2: evaluationOutputs

Bus Object for Output Port 3: vehicleOutputs

Also, the above mentioned MATLAB command script is provided as m file to the challengers.

After finishing the Step1 and Step2, you are able to run the Simulink model normally until the MATLAB session will be terminated.

C. Simulation results with example controller

Fig. 10 shows the simulation results of the Problem 1 by Simulink model using the default controller. Fig. 11 shows the simulation results of the Problem 2 by Simulink model using the default controller.

V. Provided files to the challengers

A zip file (*CDC2023_Simulink.zip*) are provided for the challengers, including following modules for Simulink environment, shown in Table V. Moreover, a Modelica source file named *CDC2023.mo* shown in Fig. 14 is provided for Modelica environment.

VI. Conclusion

A benchmark problem about the vehicle dynamics control of 4 IWM EV were introduced. Modelica model and Simulink model equipped with FMU exported from the Modelica model for each problem tasks are developed and provided to the challengers with the free license of the Modelica tool. Details of the benchmark problem definition, contents of the Modelica full vehicle model and the Simulink models were described in this article. It is highly expected that many challengers will tackle this benchmark problem.

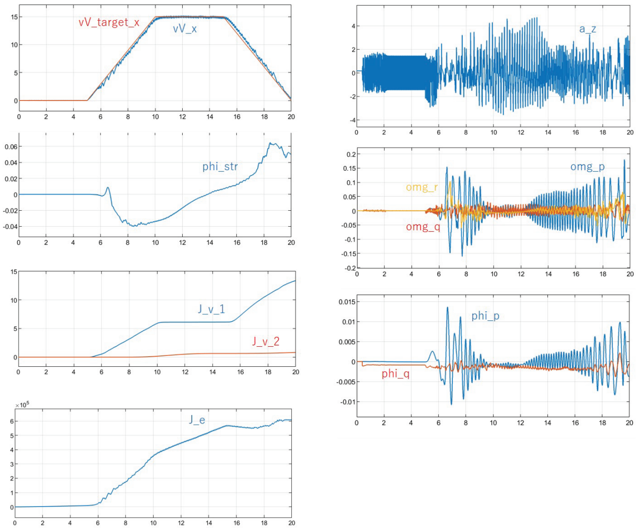


Fig. 10. Simulation results of the Problem 1 by Simulink model with default controller.

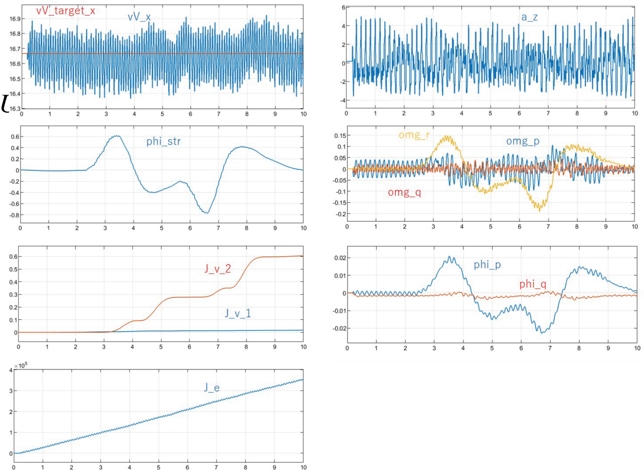


Fig. 11. Simulation results of the Problem 2 by Simulink model with default controller.

References

[1] E. Katsuyama, Decoupled 3D moment control for vehicle motion using In-Wheel motors, *SAE Int. J. Passeng. Cars - Mech. Syst.*, vol. 6(1): pp. 137-146, 2013.DOI: https://doi.org/10.4271/2013-01-0679

[2] https://modelica.org/index.html

[3] https://modelon.com/

[4] E. V. O. J. Kuiper, J. J. M. Van Oosten, The PAC2002 advanced handling tire model, *Vehicle system dynamics*, vol. 45(S1), pp. 153-167, 2007.

Appendix

A. Driver Behavior Model

Fig. 12 shows the details of the driver behavior model. Here, the 'lateralPlanning' model calculates information about points on desired path to be traced for the lateral tracker model. The 'planning' and 'planning2' models calculate path point information for different preview time and preview distance of the driver model respectively to be used by the longitudinal tracker model. In all of the planning models, the preview time and the preview distance are given as functions of the vehicle speed. The ‘percept’ model provides the information about vehicle speed to each model. The ‘lateralTracker’ model calculates the steering command to track the desired path. The ‘longitudinalTracker’ model calculates the break and accel commands to track the desired vehicle speed.

The driver model only provides the information about the desired vehicle velocity along the path and the displacement of vehicle position in x-axis and y-axis along the path as shown in Table II.

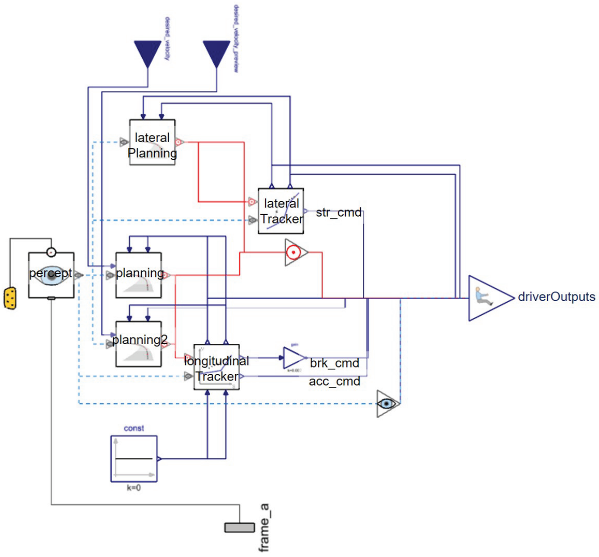


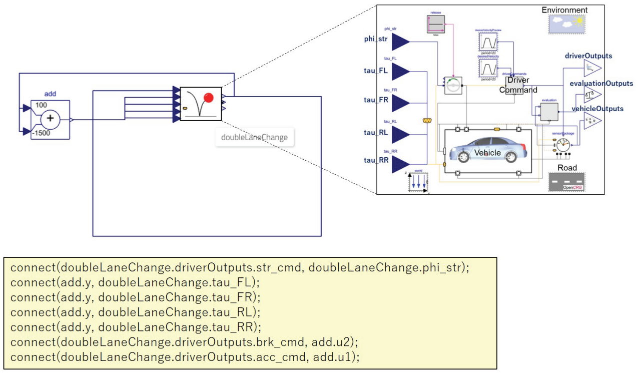
Fig. 12. Details of driver behavior model.

B. Default controller by Modelica

Default controller was made as a sample of the controller model. Torque command of each wheel and the steering command are calculated to track the desired vehicle speed and the desired course coordinates.

Fig. 13 shows an example of Modelica model of full vehicle simulation with the default controller mentioned above. The model entity ‘doubleLaneChange’ is a full vehicle model for which parameters and data were settled to simulate the double lane change test (Problem 2). As shown in Fig. 13, user can access and connect each signal by using dot-notation of the connector signal name and ‘connect’ command of Modelica grammar. (Usually, those codes are generated automatically when you connect the signals in GUI editor of Modeilca tool.) It is possible to make controller models by the similar way of connecting block models as Simulink in Modelica environment by using Modelica Block Library (Modelica.Blocks). Modelica.Blocks library is provided with each Modelica tool as a part of the free Modelica Standard Library.

Fig. 14 shows the hierarchy of Modelica library package named ‘CDC2023’ which is provided to the challengers. The full vehicle models for the Problem 1 (Acceleration) and for the Problem 2 (DoubleLaneChange) are built in CDC2023.Vehicles.Exports package. The total simulation models including the default controller are prepared in *CDC2023.Vehicles.Exports.Tests* package. When challengers use Modelica model for making their own controller, it is possible by copying and modifying each model in the *CDC2023.Vehicles.Exports.Tests* package. (Here, the models with the name added with the word ‘Map’ or ‘MapRoad’ are those using a table based road surface data and not used in this benchmark problem. The models used in this benchmark problem are using road surface data based on OpenCGR format.)



```
connect(doubleLaneChange.driverOutputs.str_cmd, doubleLaneChange.phi_str);
connect(add.y, doubleLaneChange.tau_FL);
connect(add.y, doubleLaneChange.tau_FR);
connect(add.y, doubleLaneChange.tau_RL);
connect(add.y, doubleLaneChange.tau_RR);
connect(doubleLaneChange.driverOutputs.brk_cmd, add.u2);
connect(doubleLaneChange.driverOutputs.acc_cmd, add.u1);
```

Fig. 13. Modelica model of full vehicle simulation with the default controller.

Fig. 15 shows simulation results by Modelica using the default controller for the Problem 1 (*CDC2023.Vehicles.Exports.Tests.Acceleration* model). Fig. 16 shows simulation results by Modelica using the default controller for the Problem 2 (*CDC2023.Vehicles.Exports.Tests.DoubleLaneChange* model).

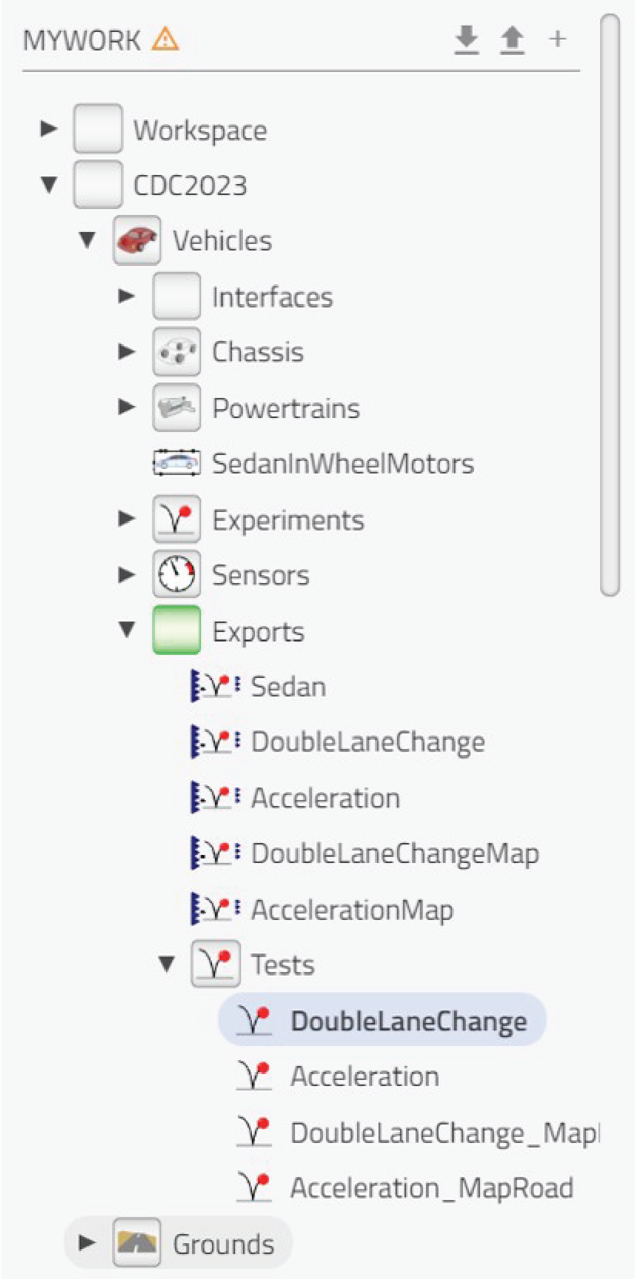
C. Procedure of Importing FMU Block

To import a FMU in Simulink, at first open the Library Browser of Simulink and then drag & drop the ‘Simulink Extras −> FMU Import −> FMU’ block into a new Simulink model. When double-clicking the new FMU block, you can specify the fmu file to be imported. For example, if you import *CDC2023.Vehicles.Exports.Acceleration.fmu*, the Simulink model will look like Fig. 17.

Next, to validate the bus objects of the FMU, execute the command below in MATLAB command window. (It is assumed that the Simulink model is saved as Acceleration_Sample.slx. Please rename the name of the Simulink model and its hierarchy in the command below.) You will get the command response as below.

fmudialog.createBusType(’Acceleration_Sample/FMU’)
Bus Object for Output Port 1: driverOutputs
Bus Object for Output Port 2: evaluationOutputs
Bus Object for Output Port 3: vehicleOutputs

After this, you will be able to extract signals from the bus connector of the FMU. You should use ‘Simulink −> Signal Routing −> Bus Selector’ block to extract necessary signals. After connecting of the Bus Selector to the FMU



MYWORK

- Workspace
- CDC2023
  - Vehicles
    - Interfaces
    - Chassis
    - Powertrains
    - SedanInWheelMotors
    - Experiments
    - Sensors
    - Exports
      - Sedan
      - DoubleLaneChange
      - Acceleration
      - DoubleLaneChangeMap
      - AccelerationMap
    - Tests
      - DoubleLaneChange
      - Acceleration
      - DoubleLaneChange_MapI
      - Acceleration_MapRoad
  - Grounds

Fig. 14. Modelica library for CDC2023 Benchmark Problem.

block and double-clicking the Bus Selector block, you can specify the signals to be extracted as shown in Fig. 18.

Please note that it is necessary to validate the bus object of FMU every time you start using the Simulink model including FMU. Also, please note that it usually takes several minutes to import and validate the FMU in the Simulink model at the first time of the Matlab session.

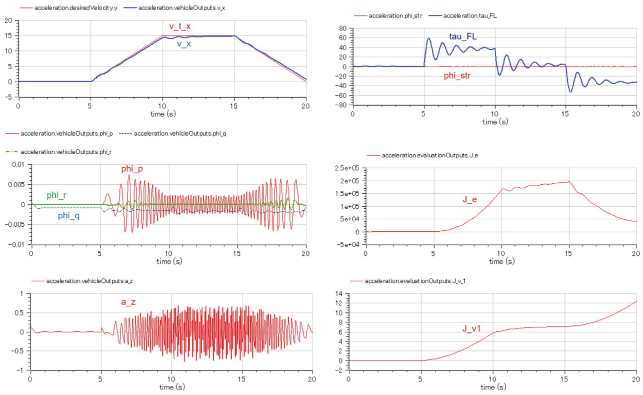


Fig. 15. Simulation results by Modelica using the default controller for the Problem 1.

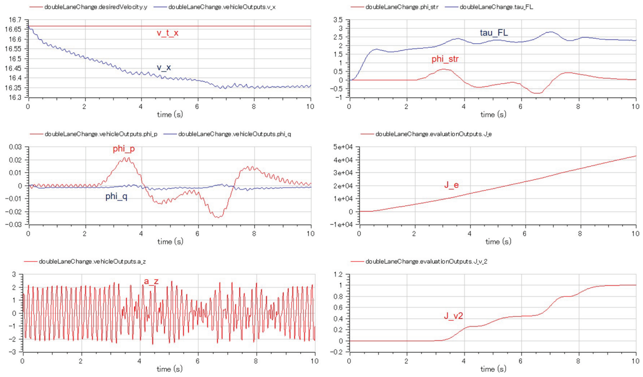


Fig. 16. Simulation results by Modelica using the default controller for the Problem 2.

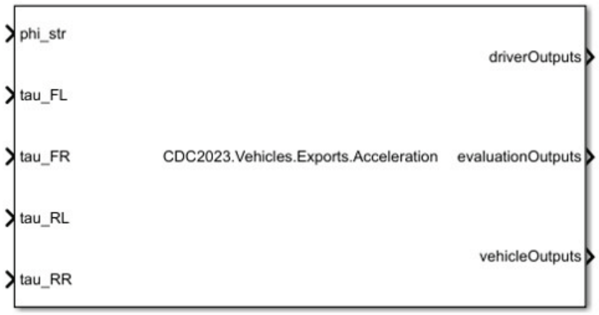


Fig. 17. Example of an imported FMU in Simulink.

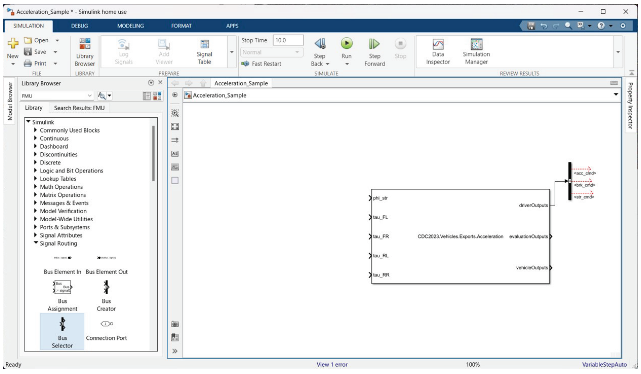


Fig. 18. Example of extracting signals from FMU connector.

TABLE IV

Definition of ‘vehicleOutputs’ connector signals

Meaning	Signal name [Units]
Body roll angle	phi_p [rad]
Body pitch angle	phi_q [rad]
Body yaw angle	phi_r [rad]
Body roll speed	omg_p [rad/s]
Body pitch speed	omg_q [rad/s]
Body yaw speed	omg_r [rad/s]
Body vertical acceleration	a_z [m/s2]
Body lateral acceleration	a_y [m/s2]
Body longitudinal acceleration	a_x [m/s2]
Body vertical speed	v_z [m/s]
Body lateral speed	v_y [m/s]
Body longitudinal speed	v_x [m/s]
Target body lateral speed	v_t_y [m/s]
Target body longitudinal speed	v_t_x [m/s]
Target body lateral position	s_t_y [m/s]
Target body longitudinal position	s_t_x [m/s]
FL wheel rotational speed	omg_FL [rad/s]
FR wheel rotational speed	omg_FR [rad/s]
RL wheel rotational speed	omg_RL [rad/s]
RR wheel rotational speed	omg_RR [rad/s]
FL wheel slip ratio	sr_FL [-]
FR wheel slip ratio	sr_FR [-]
RL wheel slip ratio	sr_RL [-]
RR wheel slip ratio	sr_RR [-]
FL wheel motor current	i_FL [A]
FR wheel motor current	i_FR [A]
RL wheel motor current	i_RL [A]
RR wheel motor current	i_RR [A]
FL wheel motor torque	tau_FL [Nm]
FR wheel motor torque	tau_FR [Nm]
RL wheel motor torque	tau_RL [Nm]
RR wheel motor torque	tau_RR [Nm]
FL wheel motor power	pow_FL [J/s]
FR wheel motor power	pow_FR [J/s]
RL wheel motor power	pow_RL [J/s]
RR wheel motor power	pow_RR [J/s]
Battery voltage	v_bat [V]
FL tyre steering angle	phi_str_FL [rad]
FR tyre steering angle	phi_str_FR [rad]
FL tyre vertical force	f_z_FL [N]
FR tyre vertical force	f_z_FR [N]
RL tyre vertical force	f_z_RL [N]
RR tyre vertical force	f_z_RR [N]
FL tyre lateral force	f_y_FL [N]
FR tyre lateral force	f_y_FR [N]
RL tyre lateral force	f_y_RL [N]
RR tyre lateral force	f_y_RR [N]
FL tyre longinudinal force	f_x_FL [N]
FR tyre longinudinal force	f_x_FR [N]
RL tyre longinudinal force	f_x_RL [N]
RR tyre longinudinal force	f_x_RR [N]

TABLE V

List of modules in Zip file

File name	File meaning
CDC_Task1_Sample.slx	Simulink model with default controller for the Problem 1
CDC_Task2_Sample.slx	Simulink model with default controller for the Problem 2
CDC2023_Vehicles_Exports_Acceleration.fmu	FMU for the Problem 1
CDC2023_Vehicles_Exports_DoubleLaneChange.fmu	FMU for the Problem 2
start_Task1.m	Matlab script at the starting used for the Problem 1
start_Task2.m	Matlab script at the starting used for the Problem 2