Application of FMI in Metal Casting press-Forming Process

LI Sheng[1,a] Chen Min[1, b] Yu Liu[1, c] Jiang Chen[1, d]

[1]CISDI R&D Co.,Ltd,ChongQing. No.11 Huijin Road, New North Zone, Chongqing, 401122,China

[a]Sheng.A.Li@ cisdi.com.cn, [b]Min.Chen@cisdi.com.cn,[c]Yu.Liu@cisdi.com.cn,

[d]Jiang.Chen@cisdi.com.cn

Abstract:*Accurate simulation of metal casting press-forming process needs to consider mutual coupling effects in a number of different fields of physics subsystem. Hydraulic systems, control systems and mechanical systems are the most important subsystems among them. It is difficult to create various subsystems in detail in a single modeling tools, so co-simulation technology is used to take advantage of different tools to achieve the entire physical process of system-level simulation. The paper researched the co-simulation in the Abaqus software and the Matlab software based on FMI standard, considered fully the coupling effect between different system, simulated the metal casting press-forming process. The simulation results showed that co-simulation based on FMI standard can be well suited for multi-disciplinary co-simulation in complex mechanical model, and played a well-guiding role in the engineering design. The co-simulation would take more computation time than traditional simulation, but it can be achieved to research the integrated features of system and to reduce greatly experiments costs and prototype trial risks by using this technology.*

Key words:*FMI&FMU; Soft Reduction; Co-Simulation; Multi-Subject Coupling*

1 Introduction

Modeling metal casting press-forming process very often necessitates modeling in multiple physical domains. The Matlab Software or the Modelica language is a natural choice for such multi-domain modeling, but for situations in which very high accuracy is required, or where physically-derived models are not available, other tools such as finite-element modeling may be more appropriate[1]. In such situations, co-simulation may be used to combine the advantages of different modeling tools to achieve the desired level of accuracy. In this paper, co-simulation between Matlab and Abaqus based the FMI standard is used to investigate the effects of soft reduction with a highly detailed finite-element Mechanical model.

The soft reduction is that a pressure which is applied near the end of liquid core of continuous casting slab produces a certain amount of reduction for compensating solidification shrinkage of the casting slab. On the one hand, this process can eliminate or reduce the closed pore formed by slab shrinkage, and prevent the flowing of liquid steel with intergranular enrichment of solute elements. On the other hand, the squeezing action caused by the soft reduction can prompt the liquid steel in the liquid core to flow reverse along the casting direction, this makes the solute elements redistribution in liquid steel and makes the solidified structure of casting slab more uniform density to improve the center segregation and to reduce the center loose. The soft reduction process is shown in the following figure 1: the two rollers reduces the slab from the two directions, the slab is rolled to the right direction, the liquid steel with intergranular enrichment of solute elements is squeezed from right to the left.

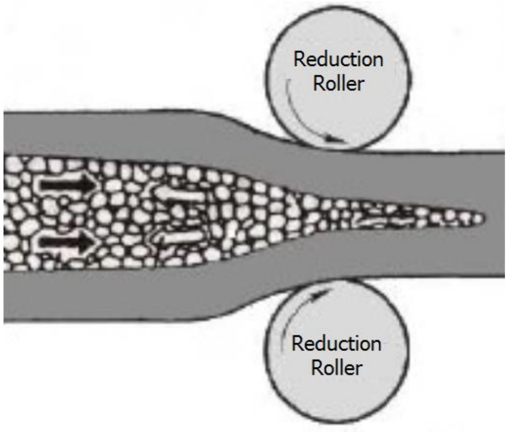

Figure 1: The principle diagram of the roller soft reduction

In the investigation, a slab with temperature field in the soft reduction system is modeled by the finite-element method in Abaqus. Matlab is used for the implementation of soft reduction system hydraulics model as well as control algorithm, then FMI standard is used to transform this hydraulics control model into the FMU model, finally Abaqus is used for the soft reduction mechanical model. The signals necessary for hydraulics reducing system are passed from Abaqus to FMU, and the FMU model provides a hydraulics reducing force output to Abaqus.

2 Hydraulics Reducing System Modeling in Matlab

In the co-simulation, the hydraulic reducing system as well as the control algorithm is implemented in Matlab. Hydraulic components from the SimHydraulics library are used. The hydraulic reducing system consists of four reducing cylinder which pull the bottom rollers and push the up rollers, and each cylinder connected to a three-position four-way valve. The valve is set up to have two modes of operation: a reducing mode in which the cylinder presses the slab, and a return mode in which the cylinder returns the initial state. Therefore, there are four feedback hydraulic control subsystems for the four cylinders. The following figure 2 shows one of the four subsystems in order to simple narrative [2].

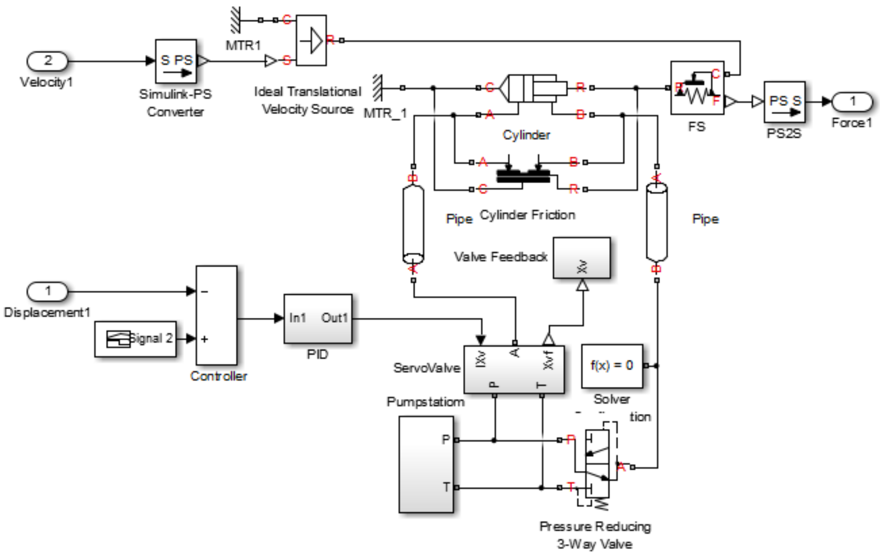

Figure 2: Diagram view of one of hydraulics control subsystem in the Matlab/Simulink

2.1 Control Algorithm

In the investigation, the preplanned reducing displacement curve along with time t is described by the following formula and the Figure 3. The period in the t<0.48s is for building stability simulation, and next for displacement tracking control.

$$\mathbf{x(t)} = \begin{cases} \mathbf{0}, & \boldsymbol{t < 0.48s} \\ \mathbf{cos(2\pi t - 0.96\pi)}, & \boldsymbol{t \geq 0.48s} \end{cases}$$

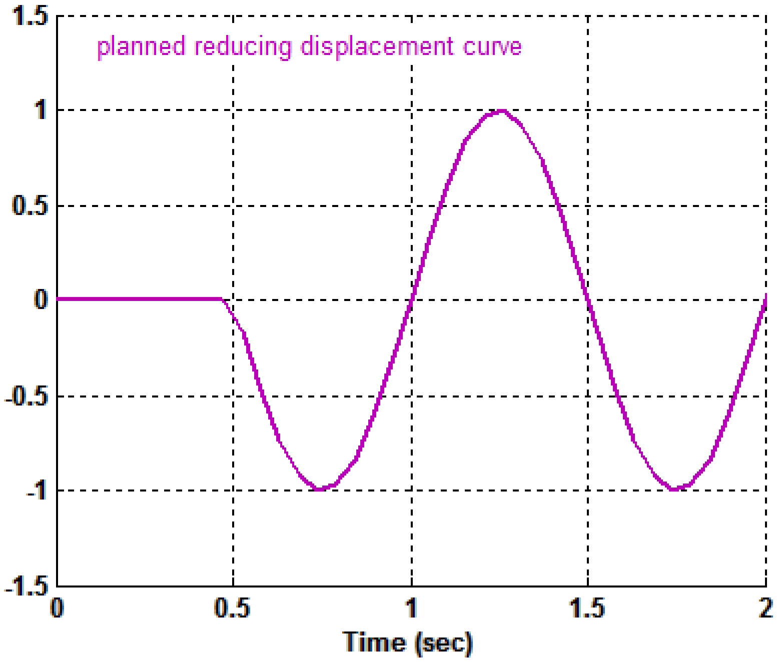

Figure 3: The curve of preplanned reducing displacement

The Control process is performed by modulating the position of the three-position four-way valve depending on the reducing status. The required inputs to the controller are reducing displacement **x**, and reducing velocity $\dot{\mathbf{x}}$,The outputs from the controller is cylinder force **F** that driving the reducing process. If the displacement difference between the preplanned position and the realistic position is less than 0, the cylinder force is less than 0, and vice versa. This feedback control process will make that the realistic displacement always follow the preplanned displacement. The input and output signals between the controller and Abaqus are sampled with period **dt** = 0.2 ms(milliseconds).

3 FMU Model from the Matlab Software

Functional Mock-up Interface (FMI) is a tool independent standard to support both model exchange and co-simulation of dynamic models using a combination of xml-files and compiled C-code. FMI can transforms the model from some specific simulation software into a Functional Mock-up Unit(FMU) model. The FMI&FMU standard is currently supported by 78 tools,such as Matlab, Abaqus, Dymola, Adams etc[3].

Before transformed into the FMU model, the control model in Matlab/Simulink needs some preparatory work, for it usually does not conforms with the FMI Characteristics. Firstly, the solver type of the control model must be fixed-step in the configuration parameters, because the co-simulation usually needs to exchange signals between different subsystems at a certain sample time. And then, all of the I/O signals Should be set as the monodirectional numerical ports instead of the bidirectional physical modeling connection ports. Finally, an appropriate fundamental sample time should be set depending on the natural frequency , driving frequency and so on. In this hydraulic control model, the solver type was the Fixed-step and the Solver was ode3, the I/O ports were $\mathbf{x}_i/\dot{\mathbf{x}}_i/\mathbf{F}_i$ (i=1,2,3,4),the fundamental sample time was 0.2 ms.

After the preparatory work, The hydraulic control model can be transformed as the FMU model. In the configuration parameters, the system target file adopted the rtwsfcnfmi.tlc from the Dymola software, the FMI version was 1.0 and the FMI type was Co-Simulation. Then the FMU model named hydcontrol.fmu for example was built, which is the control model to take part in co-simulation actually[4].

4 Soft reduction mechanical model in Abaqus

The soft reduction mechanical model is made up of the up housing in which a set of seven rollers mounted, the bottom housing in which a set of seven rollers mounted, the four reduction hydraulic cylinders and the slab with high temperature field. the blocks of the cylinders are fixed to the up housing, and the pistons of the cylinders are hinged to the bottom housing. When the lower chambers of the cylinders are full of high pressurized oil, the cylinders drive the up and bottom housings to reduce the steel slab and finish the process, showed in Figure 3.

All parts of the mechanical model are modeled in the finite element software Abaqus. There most important component in the Abaqus model is the detailed modeling of the slab. The slab with high temperature field, shape of a flat rectangle, is composed of solid all around and liquid in the center. The boundary between solid and liquid is two-way symmetrical b-spline curve as the yellow curve showed in the following figure 4[5].

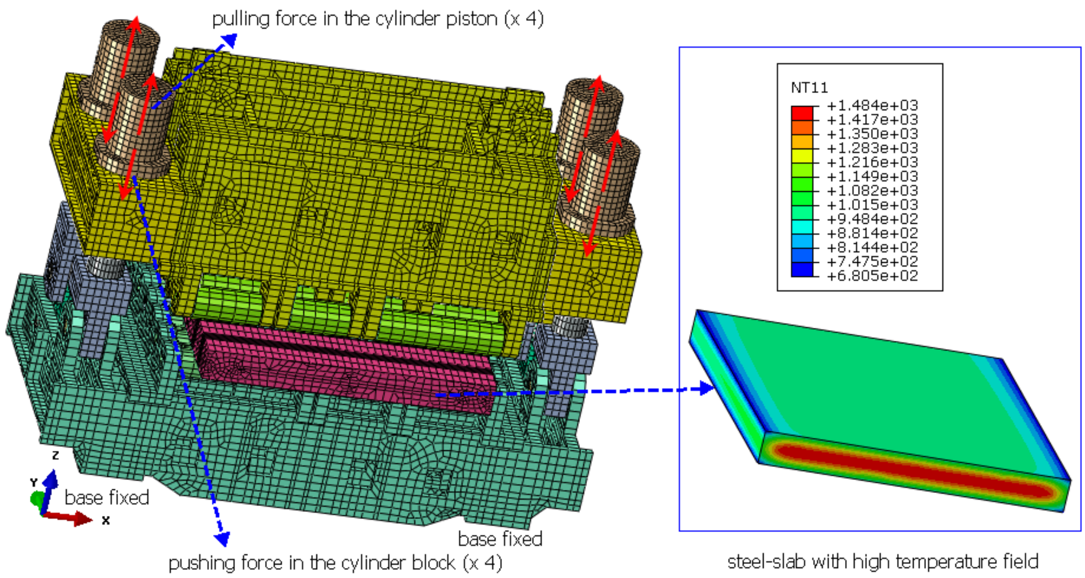

Figure 4: The soft reduction mechanical model and the slab with high temperature field in Abaqus

The reducing forces from the cylinders are then imported in Abaqus/Standard which is used for the actual reducing analysis via co-simulation with FMU. Abaqus/Standard solvers for most of the mechanics in the analysis to update the current state of reducing forces from the cylinders. Sensor information such as reducing displacement and reducing velocity are communicated at runtime at frequent simulation intervals to FMU which computes the four cylinders' forces. The forces are then communicated back to Abaqus/Standard and drive the reducing process.

The total number of degrees of freedom in the Abaqus model is more than 910000 which are basically nodal translations and rotations.

5 Simulation of Soft Reduction in Matlab

In order to test the soft reduction process in Matlab, a ideal mass-spring-damping model, representing the realistic soft reduction model, is used in Matlab. The mass value of the ideal model is the equivalent mass of the realistic model, the stiffness value of the ideal model is the synthetical stiffness in maximum reducing displacement in the realistic model, the damping coefficient of the ideal model is the comprehensive damping of the realistic model. the ideal model is constructed using components from the simMechanics library, see the following figure 5. only one part of the whole load driven by the four cylinders was modeled in order to simple narrative. The mass block was simulated with a zero initial displacement and a zero initial velocity.

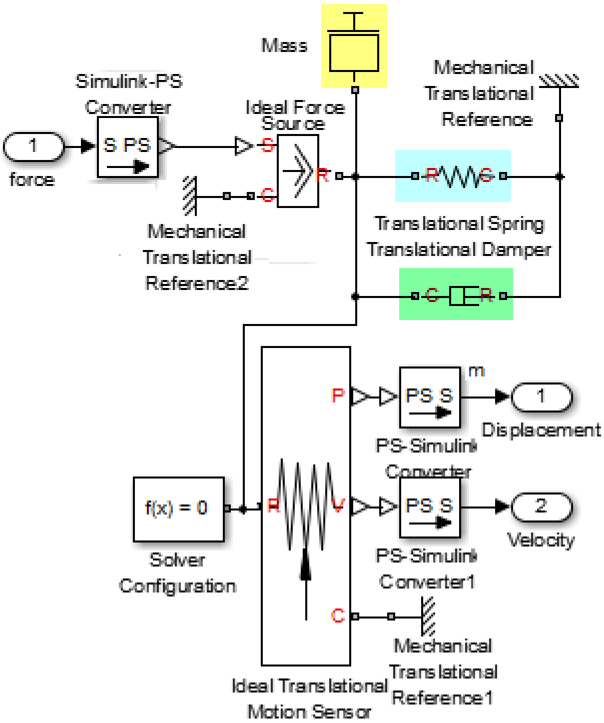

Figure 5: Ideal mass-spring-damping model

6 Co-simulation Setup

The physical system modeled with Abaqus is furnished (in Abaqus) with sensors and actuators to create what is commonly referred to as the plant. The sensor data computed by Abaqus is passed to FMU (the controller) which computes the needed actuation to drive the physical system in the desired state. The reducing displacement, reducing velocity of the blocks and pistons of the four cylinders are defined as the eight sensors in Abaqus. The four actuators(computed in FMU during co-simulation) are the four hydraulic forces to reduce the slab. The solution in Abaqus/Standard uses a difference iterative method to compute the equilibrium condition of the physical system at time **t+dt** based on the equilibrium conditions at time **t**. One of the fundamental requirements of the difference scheme is that the time step **dt** be smaller than a critical value, known as the stable time increment. In this particular application, the value of stable time increment is close to 0.5 ms. Note that this is significantly much less than the required time increment in Matlab which is 3.5 ms. The co-simulation procedure analysis requires significantly more computational resources than a standalone Matlab run. This is primarily due to the fact that finite element computations in this application are computationally very intensive given the required small time increment size and the number of solution variables involved. Consequently, given that the number of unknowns is roughly 910000 in the Abaqus model and roughly 4000 increments are required to complete the 2.0 seconds of simulation time, the computational cost of the co-simulation is dominated by far by the cost of the finite element analysis. In this particular case, the analysis was run using 10 CPUs on a HP Z800 Intel Xeon Core 2.8Ghz Windows XP 64 bit computer and required 26.6 hours to complete. By contrast, the Matlab required less than 1 minutes of actual computational time to complete on a 64 bit Windows computer using only one CPU.

The Abaqus co-simulation capabilities include various co-simulation rendezvousing schemes. But if with FMU, there are some terms should be observed[6]:

- The types of computation step in Abaqus supported by co-simulation with FMU are Dynamic/Standard and Dynamic/Explicit.
- Given the time incrementation in Abaqus requires smaller time increment than Matlab and that the required 0.2 ms time increment is smaller than the sampling frequency of sensors used in realistic soft reduction systems. In a word, the time incrementation set in Matlab, FMU or Abaqus must be consistent and less than or equal to the smallest time among them.
- There are four files to configure the whole co-simulation process. The Job.inp defines the finite element mechanical model and sets up the sensors for output signals and actuators for input signals. The control.fmu (hydraulic.fmu for example)defines the hydraulic control model and the input/output signals. The config.xml configure two models and their input-output relationship between Abaqus model and FMU model. The runCoSim.py commands and dispatch the co-simulation process.

The sensor signals from Abaqus is read in as inputs to FMU. FMU accurately integrates in time with a unified time incrementation. Actuators(cylinder reducing force) computed in FMU are communicated back to Abaqus which applies this freshly computed load as the load for the next increment. The process is repeated until the simulation time is exhausted.

The co-simulation process was also run with an even smaller time increment (0.1 or 0.05 ms) to check whether the results change. The results were virtually identical.

7 Co-simulation Results

In this section the results of the co-simulation using are presented, and compared to the results obtained using the ideal mass-spring-damping model in Matlab only.

The realistic soft reduction process was modeled in Abaqus, which involved the component deformation, large displacement, kinematic pair, contact and plastic nonlinearity, thermal mechanical coupling. By contrast, this complex model was abstracted into a simple mass-spring-damping system in the Matlab. Therefore, due to the extremely large differences in complexity of the soft reduction models in Matlab and Abaqus, it was not expected that the co-simulation results would be quantitatively very similar. And in this paper, the communication signals in the co-simulation process, such as the reducing displacement $\mathbf{x}$, reducing velocity $\dot{\mathbf{x}}$ and reducing force $\mathbf{F}$ was selected for comparison and analysis.

7.1 Comparison between co-simulation and Matlab

Figure 6 showed the soft reducing displacement $\mathbf{x}$ in the co-simulation and Matlab-only simulation respectively. The qualitative behavior was clearly similar, But the quantitative behavior was a little different. The realistic model involves the non-linear characteristics about contact, plasticity and thermal-mechanical coupling and leaded to a few phase and amplitude diversity. In the other hand, the displacement amplitude in the return stage was close to 1 mm according with the preplanned curve, but the amplitude in the reducing stage was close to 0.5 mm only equal to half of the preplanned one. Because the controller was not able to provide enough reducing force driving the maximum reducing displacement equal to 1mm. When the spring stiffness of the ideal model reduced 10 times, the reducing displacement was near to the preplanned curve showed in Figure 7.

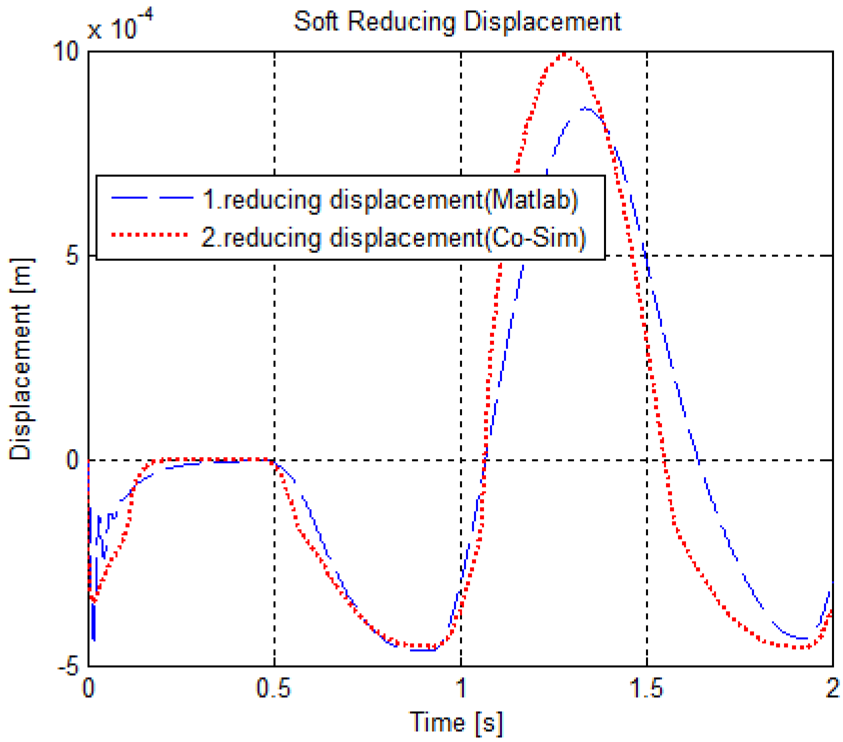

Figure 6: The soft reducing displacement curves between the co-simulation and Matlab-only simulation

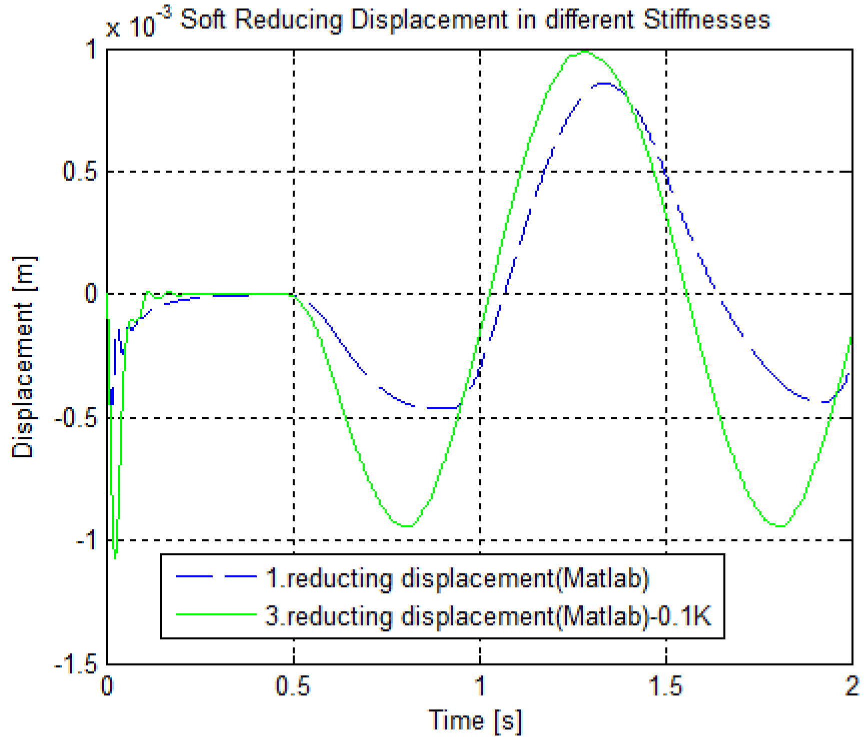

Figure 7: The soft reducing displacement curve when ideal stiffness reduced 10 times

Figure 8 showed the soft reducing velocity in the co-simulation and Matlab-only simulation respectively. The quantitative behavior is clearly similar. Except for a little difference the value of the realistic curve existed micromutations when passing through the zero displacement line(when $\mathbf{x}=0$), for the realistic model could be considered as main mass-spring model in reducing stage ($\mathbf{x}<0$) but mass-gravity model in return stage ($\mathbf{x}>0$).

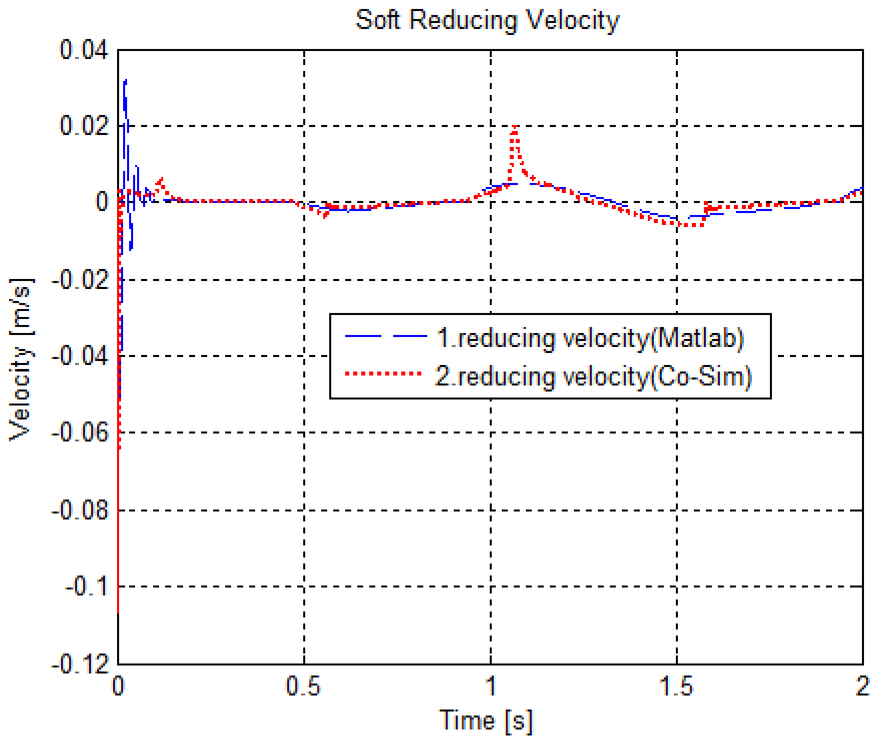

Figure 8: The soft reducing velocity curves between the co-simulation and Matlab-only simulation

Figure 9 showed the soft reducing force in the co-simulation and Matlab-only simulation respectively. The quantitative behavior was similar in reducing stage, but markedly different in return stage. Firstly, the realistic model was one-way stiffness in contrast with the two-way stiffness of the spring in the ideal model; secondly, the realistic model took only the constant gravity into account in the return stage; finally, the stiffness in the ideal model was the synthetical stiffness in maximum reducing displacement and was constant, so the peak values of reducing force was consistent, but the reducing force curves was not consistent on account of the variable stiffness when slab in elastic-plastic deformation.

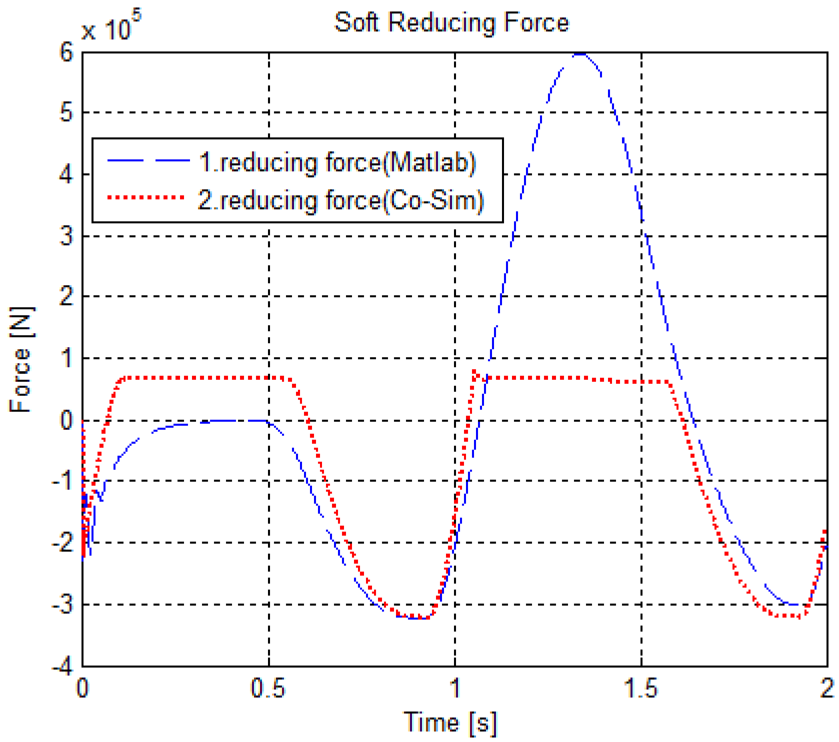

Figure 9: The soft reducing force curves between the co-simulation and Matlab-only simulation

The three main physical quantity, such as force , displacement and velocity of soft reduction, are the most interesting data in the actual production process. The designers usually analyze the system performance from the three curves and the field workers usually adjust the three variables to finish the soft reduction perfectly. For this example, the engineer could obtain the system stiffness (probably 7.1e8 N/m)and the dynamic reduction characteristics, such as the response time, stabilization precision and so on.

8 Conclusions

In this article, co-simulation was used to study soft reduction control using a high-fidelity soft reduction model. Matlab was used to implement a realistic hydraulic feedback control model and algorithm. The FMU model was transformed from the Matlab model. Abaqus was used to implement the realistic complex soft reduction mechanical model. The co-simulation was finished between Matlab and Abaqus based on FMU.

The results of co-simulation based on the FMU illuminated greatly the coupling among the mechanical-hydraulic-control system, for considering the real characteristics of the system. Through the co-simulation, the other system results, such as stress distribution, roller-gap changing, can be obtained. And many problems in variable working conditions could be found and be solved before production. As this was a preliminary investigation, there is considerable scope for future work. In particular, co-simulation with soft reduction model under the condition of the slab with walking speed or under the hydraulic cylinder synchronization control is more consistent to practical conditions. In addition, co-simulation with similar process such as rolling, tempering and straightening is also preplanned. Successful co-simulation would then allow high-fidelity simulations of similar dynamics controllers to be performed.

References

[1] Brad Schofield, Harish Surendranath, Magnus Gäfvert, Victor Oancea. Interfacing Abaqus with Dymola: A High Fidelity Anti-Lock Brake System Simulation[C]. Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009:833-838.

[2] WangYu, ZhangWenbin, LiuXun. ModelingandSimulationAnalysisfortheDynamicSoftReduction HydraulicServoSystemofaSlabContinuousCaste[J]. Mechanical Science and Technology for Aerospace Engineering. 2010, 29(3):342~345

[3] www.fmi-standard.org. Brief Introduction of the FMU.

[4] Dassault Systèmes. FMU Export from Simulink, 2015. version 2.1.

[5] FENG Ke, HAN Zhiwei, CAO Jianfeng, KONG Yiwen, WANG Shuigen. The Finite Element Analysis of the Additional Withdrawing Force Induced by Slab Soft-reduction[J]. INDUSTRIAL HEATING. 2013, 42(1):36,37,41.

[6] Dassault Systèmes. Use of Functional Mockup Unit files (FMUs) with the SIMULIA Co-Simulation Engine.2005.