

# Energy balance based Verification for Model Based Development

Kenji Sawada<sup>1</sup> Mamoru Sakura<sup>2</sup> Osamu Kaneko<sup>3</sup> Seiichi Shin<sup>4</sup>  
Isao Matsuda<sup>5</sup> Toru Murakami<sup>6</sup>

<sup>1</sup>University of Electro-Communications, Japan, {knj.sawada<sup>1</sup>,mamoru.sakura<sup>2</sup>,o.kaneko<sup>3</sup>,seiichi.shin<sup>4</sup>}@uec.ac.jp  
GAIO Technology Co. Ltd., Japan, { matsuda.i<sup>5</sup>, murakami.t<sup>6</sup>}@gaio.co.jp

## Abstract

In model-based development (MBD), it is necessary to design a multiple physical model. Integration testing of such models is not easy because of its cross-sectional development. In this paper, we propose a new unit/integration test method “energy balance based verification” based on the law of energy conservation for MBD. The key idea is that the law of energy conservation will hold for no error models. The proposed method is composed of two diagrams. The first is a hierarchical diagram considering the type of energy. The second is an energy flow diagram based on the hierarchized diagram. Also, we develop model verification tools. Through the numerical experiments, we show that the proposed method and verification tools have the possibility of judging whether the model is normal or not. In the numerical experiments, we use a mild hybrid electric vehicle model that is developed via multiple CAE: MATLAB/Simlink®, MapleSim®, and IPG-CarMaker®.

*Keywords: MBD, Model verification, Multiple CAE*

## 1 Introduction

This paper focuses on a new model verification method for Model-Based Development (MBD) using cross-sectional tools. GAIO technology Co. Ltd. (GAIO) have pushed Model-Centered Development (MCD) which targets tool developments and services for MATLAB-based MBD and UML-based MDD (Model Driven Development) so far. Especially, MCD ver.1.0 concentrates on the controller models whose code is mainly described by MATLAB/Simulink. The next generation “MCD ver.2” targets not only the controller models but also the plant models. For example, MCD ver.2 targets co-simulation based on the plant model composed of various simulation tools such as MATLAB/Simulink, Mathematica, Maple/MapleSim, IMG-CarMaker and so on. This co-simulation results in the expansion of test area not only unit tests of Function Mockup Unit (FMU) but also integration tests of Functional Mockup Interface (FMI) (Blochwitz, Torsten, et al. 2012). So, GAIO needs to enter a new stage of model verification and have to introduce new test insights.

The accuracy of MBD depends on the accuracy of the model. Typically, the accuracy of the model is categorized by two. The first is the correctness of the program code which realizes the model. The second is the correctness of the law of physics which is realized by the model. The former has been checked by typical program verification methods including unit tests and integration tests (Shokry, Hesham, et al. 2009, Rana, Rakesh, et al. 2013). The latter focuses on that ideal simulation models realize some physical laws. “The energy balanced based verification” method (Miyamoto et al. 2014) checks the energy balance of the model according to the fact that the law of energy conservation holds for no error models. That is, if the no error model has no internal loss energy, the total energy difference between the inputs and outputs will match the stored energy. Otherwise, the law of conservation does not hold. Even if the hybridization and electrification of automobiles make system structure complexity, the law of energy conservation itself does not change. This paper introduces the energy balance based verification as a new model test concept for MCD ver.2. This is the collaborative work between GAIO and the University of Electro-Communications (UEC).

This paper introduces a prototype system that streamlines the workflow of the energy balance based verification method and is composed of two verification tools. The method checks the input-output relation of each module consisting the model to calculate the energy quantity. In other words, the model expression considering the energy relation leads to efficient energy balance check of the model. Therefore, this paper proposes a hierarchical diagram and an energy flow diagram of the model. The former divides and categorizes the model according to the law of energy conservation. The category order is system, module, function, energy. The first verification tool is related to the hierarchical diagram. The latter expresses the energy flow relationship between the divided models. The second verification tool supports the energy balance based verification via the two diagram. To verify the validity of the proposed system, we consider the mild hybrid electric vehicle (MHEV) composed of MATLAB/Simulink, MapleSim, and IMG-CarMaker.

Further, we discuss the detectability of model bug using the developer tools.

## 2 Key Idea

### 2.1 Energy calculation

Energy calculation of the energy balance based verification is based on the numerical simulation of the model. We define the simulation time by

$$T_n = \Delta t \cdot n \quad (1)$$

where  $\Delta t$  is the sampling time and  $n$  is the sampled number. The simulation solver is the fixed step one. For example, the rotational kinetic energy is given by

$$\int_0^{T_n} \tau \omega dt \quad (2)$$

where torque  $\tau$ [Nm] and rotational speed  $\omega$ [rad/sec]. The translational energy is given by

$$\int_0^{T_n} N v dt \quad (3)$$

where power  $N$ [N] and velocity  $v$ [m/sec]. The electric energy is given by

$$\int_0^{T_n} V I dt \quad (4)$$

where power voltage  $V$ [V] and current  $I$ [A]. The unit of energy is [J].

### 2.2 Energy evaluation policy

First, we consider four energy components: input energy, output energy, loss energy, and stored energy. At the time  $T_n$ , the sum of input energies for the model is

$$I_n = \sum_{k=1}^{k=M} i_{k_n} \quad (5)$$

where  $M$  is the number of input energy and  $i_{k_n}$  is the  $k$ -th entry of input energy at the time  $T_n$ . At the time  $T_n$ , the sum of output energies is

$$O_n = \sum_{k=1}^{k=M} o_{k_n} \quad (6)$$

where  $o_{k_n}$  is the  $k$ -th entry of output energy at the time  $T_n$ . At the time  $T_n$ , the sum of loss energy is

$$L_n = \sum_{k=1}^{k=M} l_{k_n} \quad (7)$$

where  $l_{k_n}$  is the  $k$ -th entry of loss energy at the time  $T_n$ . At the time  $T_n$ , the sum of stored energies is

$$C_n = \sum_{k=1}^{k=M} c_{k_n} \quad (8)$$

where  $c_{k_n}$  is the  $k$ -th entry of stored energy at the time  $T_n$ . If the model has an initial stored energy  $C_o$ , the following energy conservation law

$$C_n = C_o + I_n - L_n - O_n \quad (9)$$

holds. If the simulation solver has the numerical error, the energy error

$$e_n = C_n - (C_o + I_n - L_n - O_n) \quad (10)$$

is not equal to zero. In other words, if the acceptable error of the solver  $\varepsilon$  is guaranteed,

$$\max|e_n| < \varepsilon \quad (11)$$

holds for the model following the law of energy conservation. This is the first energy evaluation policy.

The second policy is the numerical error of the loss energy. The quantity of the loss energy is positive from the input-output energy. That is, if the acceptable error of the solver  $\varepsilon$  is guaranteed,

$$\min(l_{1_n}) > -\varepsilon, \min(l_{2_n}) > -\varepsilon, \dots, \min(l_{M_n}) > -\varepsilon \quad (12)$$

hold for all loss energy components.

The energy balance based verification the correctness of the model based on (11) and (12).

## 3 Hierarchical diagram

To calculate (11) and (12) efficiently, we introduce a hierarchical diagram. The diagram categorizes modules of the target models according to the following steps:

1. Separate electric systems, mechanical systems, and composite systems and label them S1, S2, ...
2. Separate systems into modules and label them C1, C2, ...
3. Separate modules into functions and label M1, M2, ...
4. Separate functions into energies and label E1, E2, ...

The category order is system, module, function, energy. Also, the diagram has three kinds of model information. The first is the energy calculation information in subsection 2.1. The second is the energy classification of each function in subsection 2.2. The third is the relation of connection between functions.

To show the concrete example of the hierarchical diagram, we show the MHEV model as shown in Fig. 1.

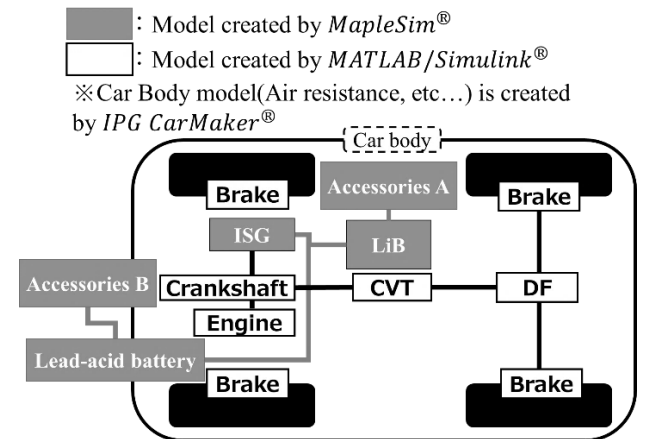


Figure 1. MHEV model.

This model is composed of MATLAB/Simlink®, MapleSim®, and IPG-CarMaker®. We apply step 1 and step 2 to the model, and we get its hierarchical diagram as shown in Fig. 2.

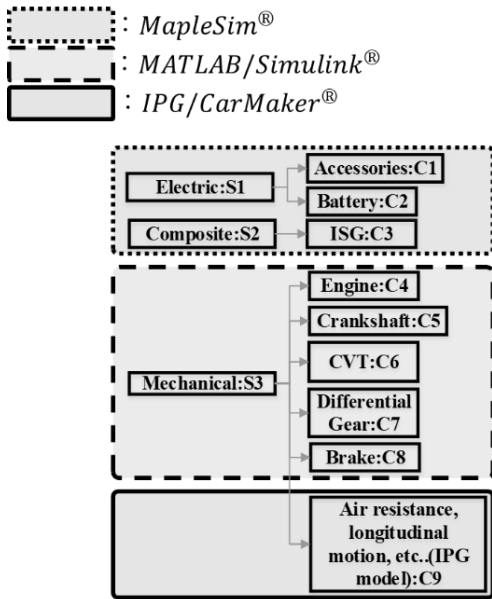


Figure 2. Hierarchical diagram of MHEV model

Fig. 3 is a part of the diagram obtained from step. 2 and step. 4.

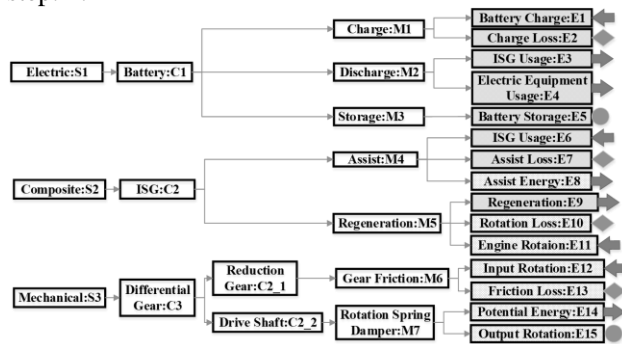


Figure 3. Detailed hierarchical diagram

In the diagram, the energy calculation information is expressed by the function block in Fig. 4.

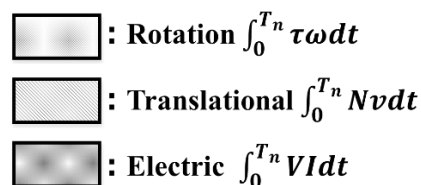


Figure 4. Relationship between energy elements and formulas

The energy classification of each function is expressed by the mark as shown in Fig. 5.



Figure 5. Energy classification in hierarchical diagram.

We design the diagram by Microsoft Visio®. Along with this, the relation of connection between functions is stored by each function block as meta information.

## 4 Energy flow diagram

The unit test of the energy balanced based verification checks (11) and (12) for each function. The integration test of the energy balanced based verification checks (11) and (12) for multiple functions. The former is corresponding to the test of FMU, and the latter is corresponding to the test of FMI. Both tests need the energy connection information. To carry out unit tests and integration tests efficiently, we introduce an energy flow (EF) diagram. The EF diagram visualizes the energy flow connection between functions and supports the energy calculations for multiple functions.

### 4.1 Diagram expression

We use a graph expression composed of place, transition, and arc in Figs. 6~10. The place is categorized into a unit test place, a loss place, an integration test place. Transition shows the testing or not. The arc shows the direction of the energy between function.



Figure 6. Unit test place



Figure 7. Loss place



Figure 8. Integration test place



Figure 9. Transition (Non-testing)



Figure 10. Transition (Testing)

We focus on C1 of Fig. 11 (deferential gear mode of MHEV) to show the EF diagram using place, transition, and arc.

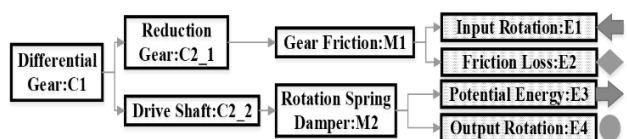


Figure 11. Hierarchy diagram of device C1

From Fig. 11, we obtain the energy relation:

$$E3_n = E3_0 + E1_n - E2_n - E4_n \quad (13)$$

Fig. 12 shows the energy flow transition of device C1. We see that the energy of the unit place changes from  $E3_0$  to  $E3_n$  and the energy of the loss place changes from 0 to  $E2_n$ . That is, we can apply (11) and (12) to each place as the unit test.

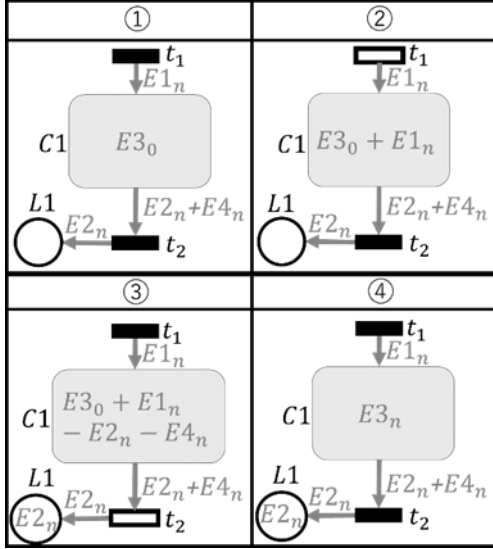


Figure 12. Energy flow transition of unit test of C1

To consider the integration test for multiple devices, we use the integration place in Fig. 8. The integration place connects unit tests as shown in Fig. 13. We consider transitions  $t_1$  and  $t_2$  as unit test A against function A, transitions  $t_3$  and  $t_4$  as unit test B against function B.  $p_1$  and  $p_2$  are for (11) and (12), and  $p_4$  is for (11) (unit test B has no loss energy). Place  $p_3$  connects function A with function B for the integration test. After unit tests A and B are carried out, the integration test checks the energy value of place  $p_3$  using (11).

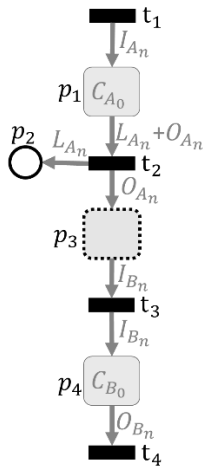


Figure 13. Energy flow between multiple functions.

## 4.2 Calculation via EF diagram

The EF diagram supports not only the visualization of unit and integration tests but also the energy calculation for (11) and (12). Consider the case of Fig. 13. The energy relation is given by

$$\begin{bmatrix} C_{A_n} \\ 0 \\ 0 \\ C_{B_n} \end{bmatrix} = \begin{bmatrix} C_{A_0} + I_{A_n} - (L_{A_n} + O_{A_n}) \\ L_{A_n} \\ O_{A_n} - I_{B_n} \\ C_{B_0} + I_{B_n} - O_{B_n} \end{bmatrix} \quad (14)$$

The 1<sup>st</sup>, 3<sup>rd</sup>, 4<sup>th</sup> low of (14) are for equation (11). The 2<sup>nd</sup> low of (14) is for equation (12), We can extract (14) from the EF diagram by introducing the mathematical graph manipulation. Considering place set  $P = \{p_1, p_2, \dots, p_\alpha\}$  and transition set  $T = \{t_1, t_2, \dots, t_\beta\}$ , we introduce the two matrices

$$\mathbf{B}^- = [b_{i,j}^-] \in \mathbb{R}^{\alpha \times \beta} \quad (15)$$

$$\mathbf{B}^+ = [b_{i,j}^+] \in \mathbb{R}^{\alpha \times \beta} \quad (16)$$

$b_{i,j}^-$  denotes the value of energy flow from place  $p_i$  to transition  $t_j$ .  $b_{i,j}^+$  denotes the value of energy flow from transition  $t_j$  to place  $p_i$ . In this case, the connection matrix of the EF diagram is given by

$$\mathbf{B} = \mathbf{B}^+ - \mathbf{B}^- \in \mathbb{R}^{\alpha \times \beta} \quad (17)$$

Also, we denote the energy values of place set  $P = \{p_1, p_2, \dots, p_\alpha\}$  as marking vector

$$\mathbf{m}_n = [m_n(p_1) \ m_n(p_2) \ \dots \ m_n(p_\alpha)]^T \quad (18)$$

and we denote the testing or non-testing information of transition set  $T = \{t_1, t_2, \dots, t_\beta\}$  as the test vector

$$\boldsymbol{\omega}_n = [\omega_n(t_1) \ \omega_n(t_2) \ \dots \ \omega_n(t_\beta)]^T \quad (19)$$

We denote  $\mathbf{m}_0$  as the initial energy values of place set  $P = \{p_1, p_2, \dots, p_\alpha\}$ , we obtain the following relation:

$$\mathbf{m}_n = \mathbf{m}_0 + \mathbf{B} \cdot \boldsymbol{\omega}_n \quad (20)$$

Equation (19) is corresponding to (13). In fact, from Fig. 14, we obtain

$$\mathbf{B}^- = \begin{bmatrix} 0 & L_{A_n} + O_{A_n} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I_{B_n} & 0 \\ 0 & 0 & 0 & O_{B_n} \end{bmatrix} \quad (21)$$

$$\mathbf{B}^+ = \begin{bmatrix} I_{A_n} & 0 & 0 & 0 \\ 0 & L_{A_n} & 0 & 0 \\ 0 & O_{A_n} & 0 & 0 \\ 0 & 0 & I_{B_n} & 0 \end{bmatrix} \quad (22)$$

$$\mathbf{m}_0 = [C_{A_0} \ 0 \ 0 \ C_{B_0}]^T \quad (23)$$

$$\mathbf{m}_n = [C_{A_n} \ 0 \ 0 \ C_{B_n}]^T \quad (24)$$

$$\boldsymbol{\omega}_n = [1 \ 1 \ 1 \ 1]^T \quad (25)$$

The matrix allows us to judge which low is for (11) or (12). If some row has all zero elements, the row is for the loss place and we apply (12) the corresponding row of (20). This expression is based on Petri net modeling (Murata 1989, Peterson 1981).

## 5 Numerical verification

The verification procedure based on energy balance is as follows:

- I. Construct the hierarchical diagram of the target model.
- II. Energy calculation via the simulation.
- III. Construct the EF diagram based on the hierarchical diagram.
- IV. Construct (11) and (12) based on (20).
- V. Carry out the unit and integration tests.

We developed the prototype system automatically carries out from Step III to Step V: Tool  $\alpha$  and Tool  $\beta$ . Tool  $\alpha$  supports Step III and Tool  $\beta$  supports Step IV and V. We apply the tools to the MHEV model in Fig. 1. Fig. 14 shows the data flow between tools.

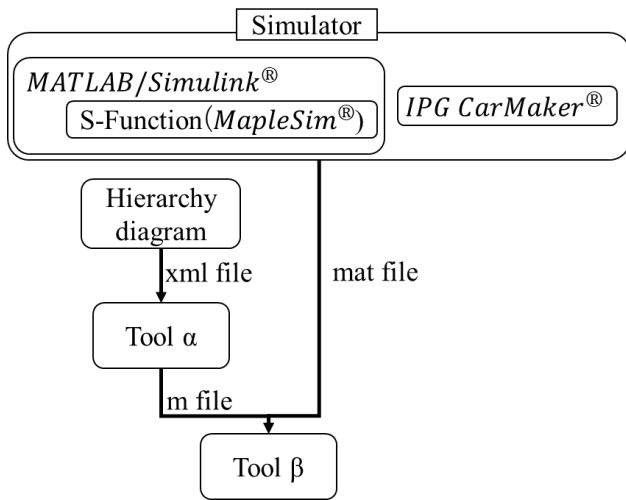


Figure 14. Data flow between tools.

In Step I, we construct the hierarchical model using Microsoft Visio to send XML files to Tool  $\alpha$ . Tool  $\alpha$  generates the connection matrix of the EF diagram for (20) and outputs m file of MATLAB. In Step II, The model of MATLAB/Simulink and the model of IPG CarMaker collaboratively via FMI. The simulation data is stored by mat file of MATLAB. Tool  $\beta$  receives m file and mat file and carries out unit and integration tests on MATLAB.

### 5.1 Verification

We consider a verification case where the MHEV model follows the energy conservation law. In Step II, we need to embed the energy calculation block in the target Simulink block as shown in Fig. 15. Fig. 16 shows the result of Tool  $\beta$ . If the model has no error, the tool outputs that the model follows the law of energy conservation.

Now, we are developing the auto-generation tool of the hierarchical diagram from the simulation model. For example, the Simulink model has XML data structure and then we can make effective use of XML. In this case,

we embed energy classification information in the tag data of the Simulink file instead of embedding the energy calculation block.

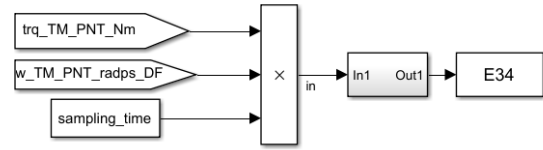


Figure 15. Energy calculation using Simulink.

```

    コマンドウィンドウ
    >> tool_b_ver2

    kekka_result =

    "It can be judged that the model is created according to the law of energy conservation"
    
```

Figure 16. Tool  $\beta$  result.

### 5.2 Bug detection

We consider the detectability of model bug using the developer tools. The model bugs of this paper are a mathematical bug and parameter bug. The former is that the equation expressing the model is wrong. The latter is that the parameter setting of the model is unrealistic. We consider the model bug of differential gear in Fig. 11.

First, we consider a mathematical bag. The following relation holds

$$G \cdot \tau_{in} = \tau_{out} \quad (26)$$

for gear ratio  $G$ , input torque  $\tau_{in}$  and output torque  $\tau_{out}$ . Also, we have the relation

$$\omega_{in} = G \cdot \omega_{out} \quad (27)$$

for input rotational speed  $\omega_{in}$  and output rotational speed  $\omega_{out}$ . Intentionally, we rewrite (27) as

$$G \cdot \omega_{in} = \omega_{out}. \quad (28)$$

This bug causes that the output energy of deferential gear is bigger than the input energy and the model does not follow the law of energy conservation. Fig. 17 shows that the tool detects the model error.

```

    コマンドウィンドウ
    >> tool_b_ver2

    kekka_result =

    "It can be judged that the model is "not " created according to the law of energy conservation"

    BUG_Place =

    8
    
```

Figure 17. Tool  $\beta$  result for error model.

“BUG\_Place” shows the row number of the connection matrix (17) in which the law of energy

