Hindawi
Complexity
Volume 2017, Article ID 1686230, 14 pages
https://doi.org/10.1155/2017/1686230

Research Article

Decomposition-Assisted Computational Technique Based on Surrogate Modeling for Real-Time Simulations

Nariman Fouladinejad, Nima Fouladinejad, Mohamad Kasim Abdul Jalil, and Jamaludin Mohd Taib

Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Johor, Malaysia

Correspondence should be addressed to Nariman Fouladinejad; fnariman3@live.utm.my

Received 31 July 2016; Revised 13 January 2017; Accepted 12 February 2017; Published 6 March 2017

Academic Editor: Francisco Gordillo

The development of complex simulation systems is extremely costly as it requires high computational capability and expensive hardware. As cost is one of the main issues in developing simulation components, achieving real-time simulation is challenging and it often leads to intensive computational burdens. Overcoming the computational burden in a multidisciplinary simulation system that has several subsystems is essential in producing inexpensive real-time simulation. In this paper, a surrogate-based computational framework was proposed to reduce the computational cost in a high-dimensional model while maintaining accurate simulation results. Several well-known metamodeling techniques were used in creating a global surrogate model. Decomposition approaches were also used to simplify the complexities of the system and to guide the surrogate modeling processes. In addition, a case study was provided to validate the proposed approach. A surrogate-based vehicle dynamic model (SBVDM) was developed to reduce computational delay in a real-time driving simulator. The results showed that the developed surrogate-based model was able to significantly reduce the computing costs, unlike the expensive computational model. The response time in surrogate-based simulation was considerably faster than the conventional model. Therefore, the proposed framework can be used in developing low-cost simulation systems while yielding high fidelity and fast computational output.

1. Introduction

Simulation is a well-known tool in both engineering system design and analysis. The basic concept of simulation is simple and understandable, which makes it a powerful decision-making instrument [1]. Vehicle simulations, such as driving simulation or flight simulation, contain many different types of components and design variables that have diverse interactions. The interdisciplinary coupling presents tough computational and organizational challenges in developing and running this kind of simulations. However, this computational challenge may simply reflect increased dimensionality, with analysis and design variables accumulated from all disciplines.

Generally, the main objective in developing simulations is to enhance an economical and efficient simulator [2–6]. The efficiency of a simulation depends on its ability in mimicking real word situations [6]. It should also be capable of providing reasonable fidelity in visual, auditory, and control feel cuing [5]. Furthermore, it is essential to have a real-time simulation to guarantee the efficiency of the system. This can be achieved when the computation of the system responses is not only accurate, but also timeliness. Using both high-performance computer systems and high-accuracy graphical system highly guarantees achieving real-time simulation. However, providing these kinds of system would be very costly. Considering the complicated subsystems and limited cost, achieving real-time simulation is therefore challenging, and it often leads to computational burdens. Computational burdens subsequently cause computational delay and inefficiency in the system.

Index reduction, model simplification, and constraint stabilization techniques were used in several studies to avoid computational delays in the system [7, 8]. However, simplified models were generally incompatible with high fidelity simulations [9]. Rather than simplification, numerical methods have been used in [9] to achieve high-speed vehicle dynamic simulation. Moreover, a multirate dynamic model was used in [6] to solve the problem of delays in the dynamical system. On the other hand, an interactive data-driven motion generation method based on a real-world motion sample was developed in [10, 11]. Although this technique improved the realistic nature of the simulation, it required costly hardware and had drawbacks including problems measuring dynamics.

In this research, metamodeling techniques were used to generate surrogate models, which were then used in place of computationally expensive models. Since there are numerous design variables and coupling in a complex simulation, decomposition technique was required and included in the process of surrogate modeling. A surrogate-based simulation framework that can be employed in a vehicle simulation to reduce the computational costs between the subsystems was proposed.

The present paper is organized as follows. Related works and system decomposition are, respectively, reviewed in Sections 2 and 3. In Section 4, a computational framework is presented based on decomposition and metamodeling techniques. The details for the case study on developing the surrogate-based vehicle dynamic model (SBVDM) using the proposed computational technique are provided in Section 5. Finally, Section 6 presents the conclusion.

2. Related Works

A nonlinear modeling ability of artificial neural network (ANN) was used in [12] and the dynamic behavior of a two-axle vehicle was simulated. The authors investigated the use of an ANN as an emulator for modeling the vehicle dynamics, since it can be simply used to learn and map the input-output relationship of nonlinear functions [10]. A simplified nonlinear vehicle model with a quasistatic load transfer assumption was used to obtain the training data at a wide range of speeds and steering inputs. Subsequently, these training data together with calculated outputs were used to train a recurrent neural network called a "Neuro-Vehicle."

Overall, the performance of the Neuro-Vehicle indicated that the neural networks are fast and relatively accurate in predicting the nonlinear behavior of a vehicle [10]. However, a comprehensive vehicle dynamic model was not used, given that the ultimate goal of their study was to develop an early warning system for conditions leading to vehicle rollover. Instead, they used a simplified model as a vehicle model and did not look into reducing computational burden. In practice, only nonlinear lateral vehicle dynamics were considered in their study. Several other researchers also predicted vehicle dynamics using ANN [11–15]. Actual vehicle measurements were used in [15] to model lateral vehicle dynamics using a neural network with two-stage hybrid learning for accurate long-term prediction. Backpropagation and general regression neural networks in modeling vehicle dynamics were employed in [14]. The total number of training data patterns of 4,500 was chosen, which equated to only 22 seconds of data. Only 14 parameters were included in the network training. Next, a radial basis function (RBF) network was used in [13] to identify nonlinear factors of four-wheel steering (4WS) vehicle dynamics. In addition, the RBF network was also used in hybrid modeling of vehicle dynamics simulation in [13]. The authors represented a good classification of different techniques for modeling dynamic systems.

3. Decomposition

As was discussed in the previous section, metamodeling techniques, such as ANN, are useful in reducing the computational burden in a system. These techniques have been shown to be effective for a single-discipline model with a small number of design variables [10]. However, difficulties arise when a simulation contains many different interacting components and design variables. This coupling presents both computational and organizational challenges. Nonetheless, the computational challenges may simply reflect increased dimensionality, with analysis and design variables from all involved disciplines. A decomposition approach is a good solution to this problem, since it is able to break large intractable simulation into smaller subsystems while maintaining the couplings among them [16].

The relationship between variables and outputs in a nonhierarchical system is often represented using a partitioning matrix. A nonhierarchical system has no starting point and it contains lateral couplings. Therefore, there are two-way couplings between many of the modules. In addition, in a nonhierarchical system, the output of one module is also the input to a second module and vice versa. Typically, computational analysis models are decomposed by partitioning their relationship matrices. Various techniques have been used in the literature to represent relationship matrices, such as dependency structure matrices, interaction matrices, function dependent matrices (FDMs), and design structure matrices (DSMs) [17].

In this study, DSM was used to represent the basic relationship between objects in the simulation. An ideal decomposition process would involve partitioning the relationship matrix into several perfectly independent, uncoupled blocks without any interactions between them. However, some interactions generally exist between the blocks in a realistic decomposition. Sensitivity analysis can be used to judge the importance of variables or their interaction terms [16–18]. Decomposing a high-dimensional model into smaller submodels presents several advantages; it allows conceptual simplification of large-scale models, reduces both the dimensionality of the overall model and necessary programming and debugging, and enhances the reliability and robustness of the resulting surrogate model. Thus, decomposition can be used to guide the metamodeling process. This approach simplifies a large-scale simulation by partitioning it into smaller submodels that are suitable for global surrogate modeling.

4. Global Surrogate-Based Modeling

An applicable approach to overcome difficulties in an expensive computational system is to use surrogate models in lieu of the computationally expensive subsystems. However, high-dimensional simulations typically contain many subsystems and variables, making it impractical to replace them with a single surrogate model (SM) [10, 16]. Figure 1 illustrates the general structure for developing a global surrogate-based model out of the high-dimensional simulation system. The high-dimensional model is firstly analyzed using both system analysis and sensitivity analysis (SA). These analyses are an excellent start for decomposing a computation-intensive model into smaller submodels that are relevant enough for surrogate modeling. As shown in Figure 1, several approaches, such as DSM, are used in the decomposition section to assist the partitioning process.

The goal of decomposition is to determine the number of required surrogate models, which form the global surrogate-based model (GSBM). These surrogate models, which are used to develop the surrogate-based simulation, are created in the surrogate modeling section. The steps in surrogate modeling are (i) sampling and design of experiments, (ii) creating the surrogate model after selecting the relevant metamodeling model type, and (iii) hyperparameter optimization and fitting the constructed model to the original model. Thus, considering the decomposition results, several surrogate models are constructed, which eventually form the GSBM.

The GSBM is tested in the final part of the framework to check whether it reaches the acceptable accuracy. If and when developing a surrogate-based model leads to models with poor predictive accuracy, several techniques, such as hybrid metamodeling [19], can be employed to improve the accuracy. The accuracy of the resulting model is then checked in the surrogate-based model analysis section. If the desired accuracy cannot be reached, the decomposition process is repeated and the number of required surrogate models is updated.

Subsequently, Figure 2 illustrates the general flowchart for developing a GSBM based on a computationally expensive simulation model. The said flowchart serves as a good guideline for developing surrogate models for a high-dimensional model. As depicted in Figure 2, the simulation model is firstly analyzed to identify the number of variables, discipline, and couplings in the system. According to the system analysis outcome, the simulation system is decomposed to several submodels, which are the relevant guideline for global surrogate modeling. Using the decomposition results, the number of required surrogate models is specified. Subsequently, several surrogate models are constructed to form the GSBM. Different sampling strategies [20], experimental design [21], and metamodeling techniques [22] are used to build the surrogate models. Surrogate models are fitted to the generated database gathered by sampling. Once the surrogate models are formed, they are tested for validation. Each surrogate model includes several parameters. Hyperparameter optimization is used to optimize these parameters and improve the accuracy of a single surrogate model. Afterward, the constructed surrogate models are chained together in a cascade form to create the GSBM. The performance of the GSBM is then evaluated, and hybrid metamodeling is used to improve the final accuracy.

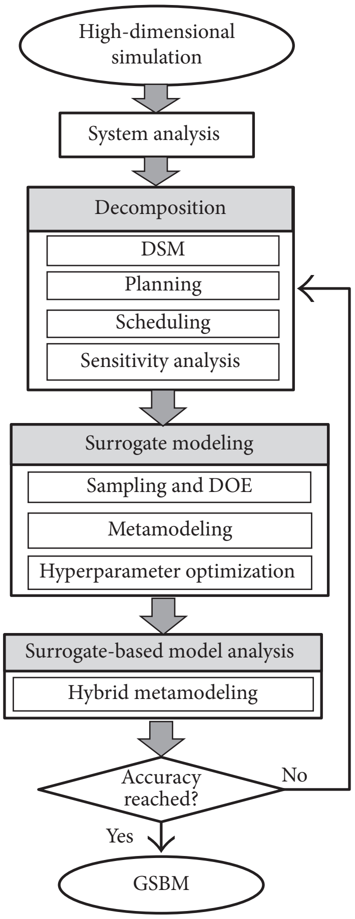


Figure 1: General structure for surrogate-based simulation development.

If the acceptable accuracy could not be reached, the decomposition process is updated and a SM is added to the area with extreme deviation. However, a high number of SMs could lead to computational burden. The process of updating the GSBM is continued and repeated until the relevant accuracy and computational cost are met. The main challenge is to determine the trade-off between the computational cost and accuracy of the surrogate-based model. This research could provide a unique methodology to fill this gap by updating a GSBM with different number of surrogate models and samples as well as various types of metamodeling techniques. The best-formed surrogate-based model should be finally validated, in terms of both accuracy and computational time, to achieve a real-time simulation.

The GSBM is then used in lieu of conventional model in the simulation. It is expected that once the surrogate-based model is used in the system, the simulation will definitely perform faster, compared to the primary source, while still being accurate enough.

In the following section, the process of creating a GSBM for a driving simulator is presented.

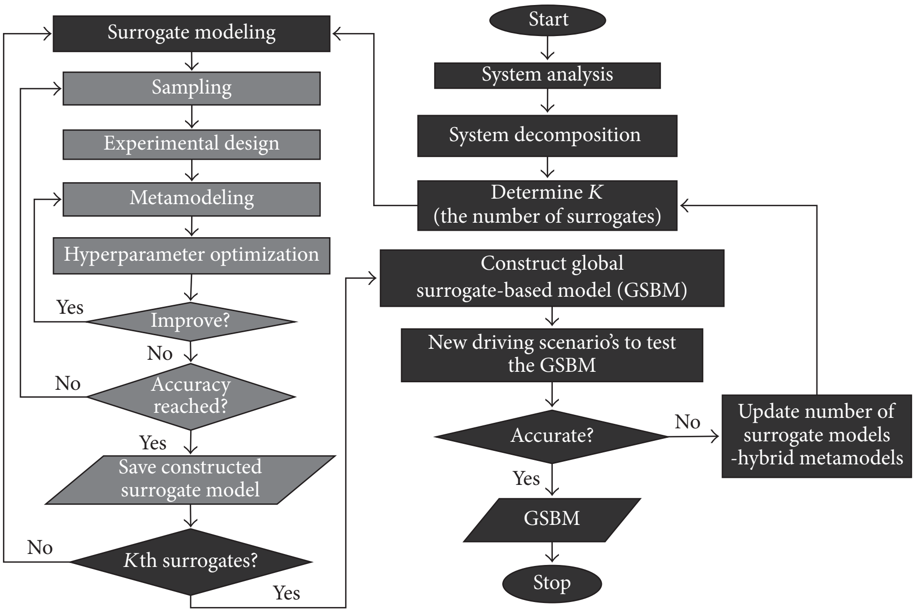

Surrogate modeling
Sampling
Experimental design
Metamodeling
Hyperparameter optimization
Yes
Improve?
No
No
Accuracy reached?
Yes
Save constructed surrogate model
No
Kth surrogates?
Yes
Start
System analysis
System decomposition
Determine K (the number of surrogates)
Construct global surrogate-based model (GSBM)
New driving scenario's to test the GSBM
Accurate?
No
Update number of surrogate models -hybrid metamodels
Yes
GSBM
Stop

Figure 2: Flowchart of developing a global surrogate-based model (GSBM).

5. Case Study: Surrogate-Based Vehicle Dynamic Model (SBVDM)

The Universiti Teknologi Malaysia driving simulator (UTMDS) was used to illustrate the proposed methodology. UTMDS consists of several subsystems, namely, the Universiti Teknologi Malaysia vehicle dynamic model (UTMVDM) calculating the motion of the simulated car [2]; the motion platform performing and transferring a vehicle motion to the driver [23]; the driving environment scene and sound reproducing virtual environment [24]; control devices acting as an interface between the operator and driving simulator; and system integration managing the communication and data transfer among the subsystems [25]. Figure 3 depicts the subsystems of UTMDS.

The ability to run the system in real-time depends on the integration time step and complexity of the vehicle dynamic model (VDM). Most of the computational burdens often happen in VDM, as it is consists of various mathematical models and subsystems. There are six main subsystems in UTMVDM including engine, powertrain, load transfer, tire, handling, and suspension models. As shown in Figure 3, UTMVDM directly interacts with the virtual driving environment (VDE). Smooth-appearing motion in the VDE is essential in UTMDS so that the human operator can anticipate vehicle movement. The visual cueing must be updated at a sufficient rate to generate an impression of smooth motion. Similarly, the transport delay must be minimized to avoid degrading the driver/vehicle bandwidth.

UTMVDM has been validated to a time step of 50 milliseconds [25]. However, a smaller time step is required to reach high fidelity and realistic simulation. In addition, there is some lag in transferring data from UTMVDM to other subsystems. Thus, computational delay is the major issue to be resolved in UTMVDM [26]. Computational delays in UTMVDM cause UTMDS to be unrealistic and reduce the system fidelity. The goal of the case study was to reduce the computational delays in UTMVDM using a surrogate-based computational technique. In the past, a few studies have been conducted to reduce the computational delay in the DS subsystems. Although some of these studies were successful in achieving real-time simulation, none of them could represent a low-cost computational technique that can reduce the response time in a comprehensive VDM.

The proposed surrogate-based simulation framework was used to reduce the computational burden in UTMVDM. The surrogate-based vehicle dynamic model (SBVDM) was developed to overcome the computational delay. The driving simulator, in which SBVDM was used, was called the surrogate-assisted driving simulator (SADS). According to the presented flowchart in Figure 2, the first step in creating SBVDM is system analysis.

5.1. System Analysis. UTMVDM is a nonhierarchical system, in which the interactions among the subsystem modules cannot be distributed in a traditional top down hierarchy (see Figure 4). Nonhierarchical systems, such as UTMVDM, are characterized by subsystem analyses coupled through transference of output data. These systems normally create a complex network, which is prohibitively time consuming due to the complexity of the underlying analysis codes. As illustrated in Figure 4, the engine model and powertrain were coupled with a two-way interaction. Powertrain, vertical load, and tire models interacted in a top down hierarchy. Both tire and handling models, however, were completely coupled. In other words, the outputs of the tire model were the inputs to the handling model, and vice versa. On the contrary, suspension had a one-way interaction with the handling model, and its outputs were not coupled with any

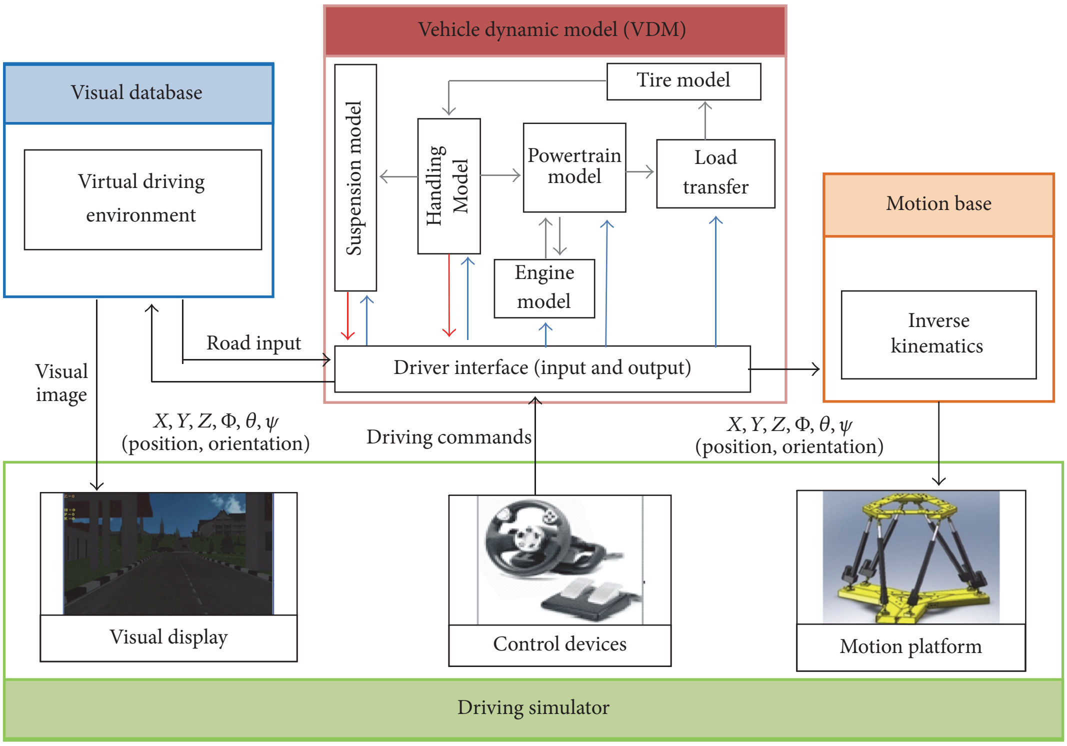


Figure 3: UTMDS subsystems.

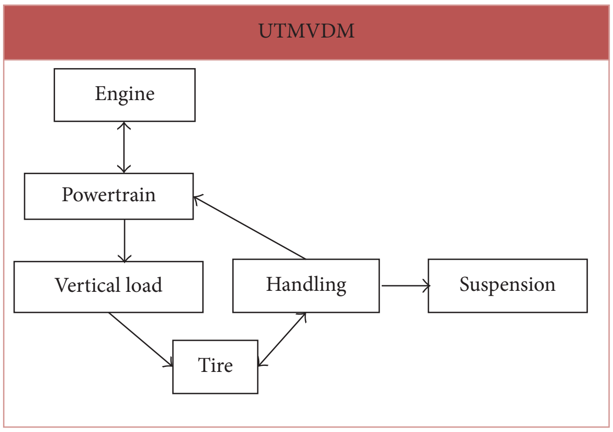


Figure 4: Interactions among the subsystems of UTMVDM.

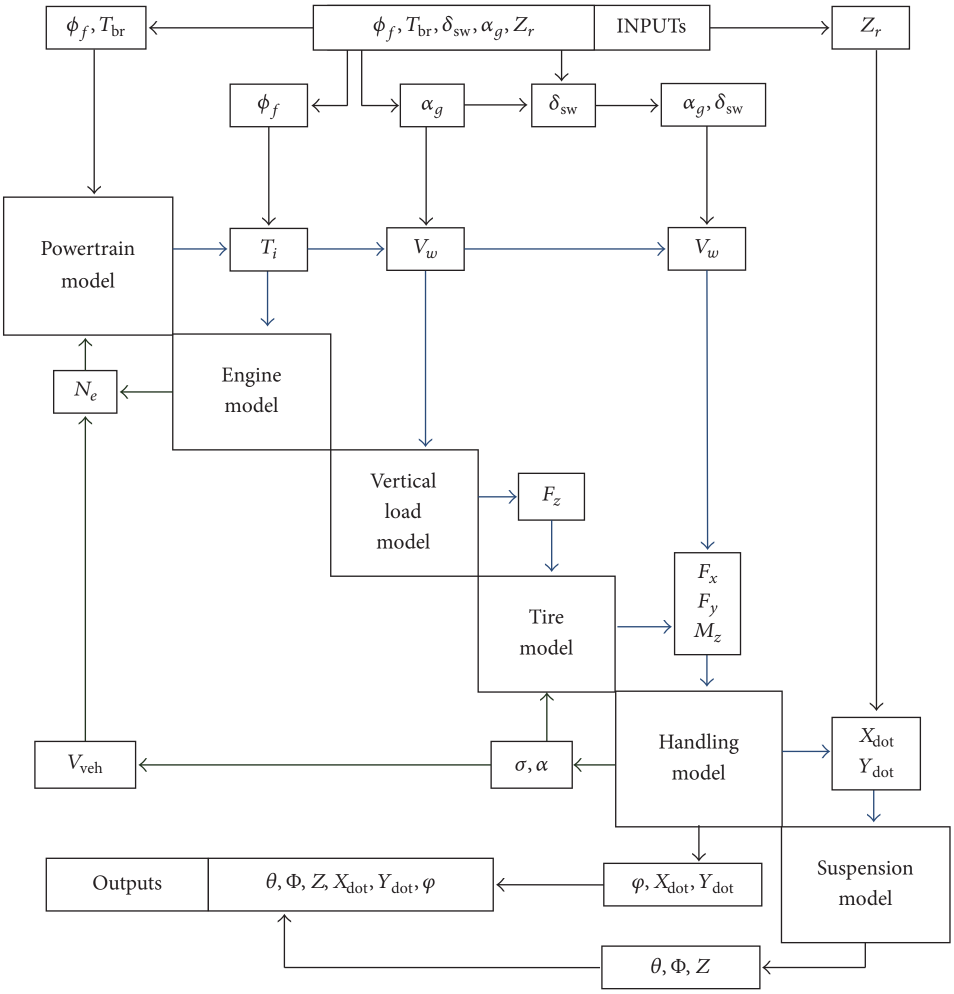
other subsystems. Each participating discipline in this system was modeled as a subsystem for which its inputs and outputs are identifiable. These inputs and outputs are clearly listed in Table 1.

The time step for a high fidelity vehicle dynamic model should be 1 millisecond to develop a real-time driving simulator [6]. As such, the computational time for UTMVDM should be less than 1 millisecond. UTMVDM computational time for 8 out of 10 runs was more than 1 millisecond, ranging between 2 and 5 milliseconds for different runs.

Table 1: Data list of inputs and outputs in the UTMVDM for decomposition.

Subsystem	Output	Input
Powertrain model	T_i, V_w	$\phi_f, T_{\mathrm{br}}, V_{\mathrm{veh}}, N_e$
Engine model	N_e	ϕ_f, T_i
Vertical load	F_z	α_g, V_w
Tire model	F_x, F_y, M_z	F_z, σ, α
Handling model	$\varphi, X_{\mathrm{dot}}, Y_{\mathrm{dot}}, \sigma, \alpha$	$F_x, F_y, M_z, \alpha_g, \delta_{\mathrm{sw}}, V_w$
Suspension model	θ, Φ, Z	$Z_r, X_{\mathrm{dot}}, Y_{\mathrm{dot}}$

5.2. UTMVDM Decomposition. The first step in decomposition was to represent the complex system as a square design structure matrix [27], wherein each of the subsystems is denoted as a box along the diagonal. Figure 5 shows the ordered design structure matrix (DSM) of the UTMVDM subsystems. Both design and dependent variables, responses, and data dependencies are clearly indicated in the said figure [18, 28]. It is important to minimize the number of feedback instances to reduce computational costs, and at the same time facilitate the decomposition process. The number of feedback instances in Figure 5 was minimized by reordering the subsystems along the diagonal. An optimally arranged system would have a hundred percentage feed forwards and no feedback instances. The process of reordering modules to improve the DSM diagram is called scheduling [16].



σ: longitudinal slip
α: sideslip
α_g: road grade
ϕ_f: throttle
δ_{sw}: wheel steer angle
θ: pitch angle
φ: yaw rate
Φ: roll
F_x: longitudinal tire forces
F_y: lateral tire forces
F_z: vertical force
M_z: moments
N_e: engine speed
T_{br}: brake torque
T_i: impeller torque
V_w: wheel speed
V_{veh}: vehicle speed
X_{dot}: longitudinal velocity
Y_{dot}: lateral velocity
Z_r: road input
Z: bounce

FIGURE 5: Ordered DSM of UTMVDM.

The suspension model had inputs from handling and outputs that did not participate in other subsystem analysis. Thus, the planning stage automatically deleted the suspension model from the design structure matrix since there was no coupling between its outputs with other subsystem. In the next step, tightly coupled subsystems, which had two-way interactions, were collapsed into a single module, termed a circuit. This process was continued to the point that the number of feedback instances was reduced to zero.

As depicted in Figure 6, UTMVDM was finally partitioned into three submodels using decomposition techniques. Figure 6 is a basic guidance to begin the surrogate modeling process. Three different surrogate models (SMs) were initially generated based on the decomposition results. The surrogate models were chained together in a cascade model, as shown in Figure 7. One surrogate model was used to replace the powertrain and engine models. The second surrogate model was used in lieu of the vertical, tire, and

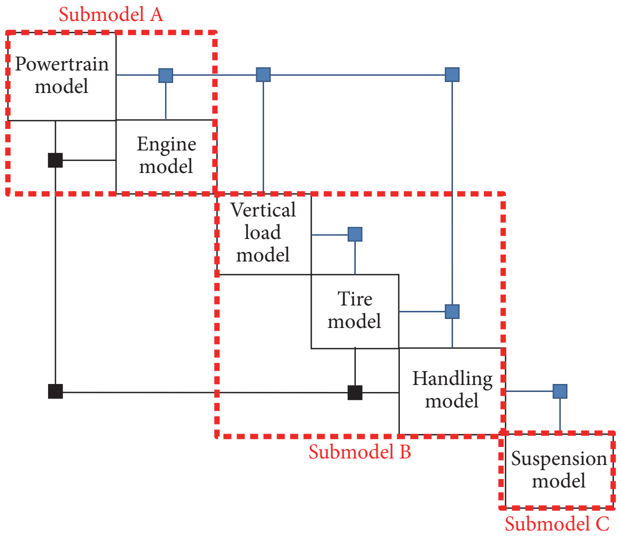


Figure 6: Partitioned UTMVDM for surrogate modeling.

handling models, while the last surrogate model was used in place of the suspension model. Figure 7 illustrates the structure of the surrogate models in developing the first version of the GSBM. The next phase in developing SBVDM was to create the surrogate models. As was previously shown in Figure 2, the first step in surrogate modeling is sampling.

5.3. Sampling. The objective of this stage of the study was to develop global surrogate-based model (GSBM) architecture, which was capable of emulating vehicle responses within a wide range of operating conditions. Thus, UTMVDM simulation was performed using a wide range of speeds, brake torque, steering, and road inputs. The roads in the virtual environment were based on a topographic map of the Universiti Teknologi Malaysia (UTM) campus [25].

Considering the speed limit on the UTM roads, the speed in the driving scenarios was limited to 0–100 km/h. In addition, the vehicle was given three types of brake torque: easy (100 N·m), moderate (200 N·m), and hard (300 N·m). Bearing in mind the maneuver types used in [10], sinusoidal steering maneuver and step change steering angle were used to train the surrogate models. The maneuvers were conducted at vehicle speeds of 10, 30, 70, and 100 km/h for sinusoidal maneuver and 100, 85, 55, and 5 km/h for step input. Several driving scenarios were required to prepare the samples to construct the surrogate models.

Six different inputs were assumed as the sampling variables. The vehicle's dynamic responses were simulated by passing speed bumps at random speeds of between 0 and 40 km/h. The range of sampling inputs is clearly illustrated in Table 2. It should be noted that each surrogate model could be constructed using a limited number of samples. Therefore, there was a limitation in selecting samples for the driving scenarios.

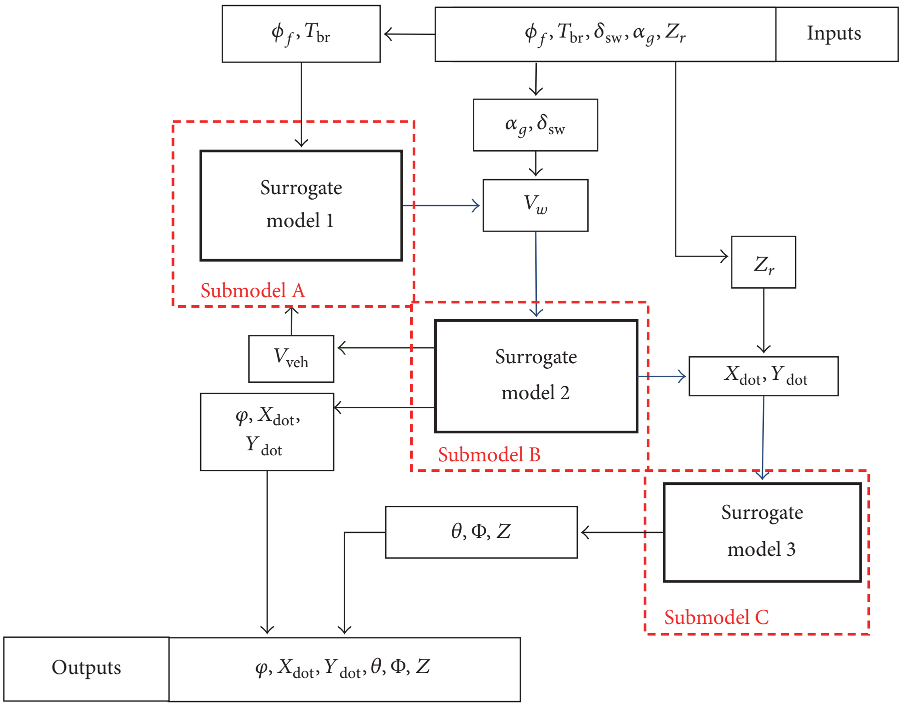
Initially, 20 different driving scenarios were used to train the surrogate models by varying the characteristics of the virtual road and simulated vehicle. The vehicle model used for the simulation was a Ford Taurus. An algorithm was used to randomly generate the said 20 driving scenarios. Each scenario lasted for 10 seconds and was conducted at time steps of 0.1, 0.01, and 0.001 seconds. Larger step sizes were used for linear areas, while smaller step sizes were used for nonlinear areas to increase the accuracy of the responses. The process of creating driving scenarios, sampling, and constructing surrogate models was done automatically. This meant that the samples were automatically selected and gathered together with the responses of UTMVDM.

5.4. Surrogate Modeling. Version 7.12.0 of Matlab, its neural network toolbox, and SUMO toolbox 7.0.2 [29] were used to create the surrogate models. A 3.10 GHZ Intel Core i5-2400 CPU Dell Vostro 260 desktop computer with 4 GB RAM was used in this study. Space filling was mainly considered in choosing sampling strategy. Latin hypercube design was mostly used as the design of experiment method. Polynomial regression [30], Kriging (KRG) [31], radial basis function (RBF) [22], multivariate adaptive regression splines (MARS) [20], and artificial neural network (ANN) [32, 33] were used as the metamodeling techniques to construct the surrogate models. Cross-validation method was employed to validate each surrogate model. The comparison among different metamodeling techniques demonstrated that polynomial model was computationally cheaper than others; however, it was not able to efficiently emulate nonlinear responses [34]. ANN, on the other hand, was found to be not as expensive and was capable of reproducing nonlinear outputs. Thus, the metamodeling stage of the surrogate-based modeling should begin with ANN [10, 35].

5.5. Performance Evaluation of the SBVDM. In developing the first version of SBVDM, the evaluation of the performance of the said model in emulating the conventional model is needed. Therefore, different samples, or in the current case study, driving scenarios, were used to check the efficiency of SBVDM. A new driving scenario was randomly selected to test SBVDM using an algorithm, which lasted for 38 seconds, and it included steering angle, road unevenness, and road grade. To evaluate the performance of the global surrogate-based modeling, the responses of SBVDM were compared against the outputs of UTMVDM. The percentage of relative error (PRE) was used to estimate the accuracy of SBVDM. PRE was calculated as follows:

$$\mathrm{PRE} = \frac{y - \hat{y}}{\max |y|} \times 100, \qquad (1)$$

where both the variables y and $\hat{y}$ represented the true and predicted responses, respectively. The maximum acceptable value for PRE was considered at 20%. Furthermore, the mean PRE should not exceed 10% to achieve acceptable accuracy for the SBVDM responses [2, 10, 34, 36]. When a large discrepancy exists between the responses of both UTMVDM and SBVDM, the surrogate-based model should be optimized. In addition, the accuracy of SBVDM could be improved by utilizing more surrogate models or hybrid metamodeling. However, it is important to note that the use


ϕ_f, T_{br}
$\phi_f, T_{\mathrm{br}}, \delta_{\mathrm{sw}}, \alpha_g, Z_r$
Inputs
$\alpha_g, \delta_{\mathrm{sw}}$
Surrogate model 1
Submodel A
V_w
Z_r
V_{veh}
Surrogate model 2
Submodel B
$X_{\mathrm{dot}}, Y_{\mathrm{dot}}$
$\varphi, X_{\mathrm{dot}}, Y_{\mathrm{dot}}$
θ, Φ, Z
Surrogate model 3
Submodel C
Outputs
$\varphi, X_{\mathrm{dot}}, Y_{\mathrm{dot}}, \theta, \Phi, Z$

α_g: road grade
ϕ_f: throttle
δ_{sw}: wheel steer angle
θ: pitch angle
φ: yaw rate
Φ: roll
T_{br}: brake torque

V_w: wheel speed
V_{veh}: vehicle speed
X_{dot}: longitudinal velocity
Y_{dot}: lateral velocity
Z_r: road input
Z: bounce

Figure 7: Initial decomposition results for the global surrogate-based modeling of UTMVDM.

Table 2: Sampling inputs and their range.

Sampling variables	Min	Max
Vehicle speed, V_{veh} (km/h)	0	100
Brake torque, T_{Br} (N·m)	0	300
Steering angle, δ_{sw} (degree)	−90	90
Road input, Z_r (m)	−0.1	0.1
Throttle, ϕ_f (%)	0	100

of more surrogate models for the creation of SBVDM would require another decomposition of the system. Therefore, the accuracy of SBVDM was evaluated in every cycle. If the desirable accuracy was not reached, the decomposition process was restarted from the beginning once more, and a new SBVDM with more surrogate models would be constructed.

As more surrogate models were added to the GSBM, the computational cost increased. Hence, the process of adding new surrogate models to SBVDM was done step by step to ensure that the results were acceptable. The computational time should be evaluated for each SBVDM to ensure that it is faster than UTMVDM. Hybrid metamodeling could also be used to increase the accuracy in areas with large deviation.

The primary decomposition results specified three surrogate models to form SBVDM (see Figure 7). These surrogate models were chained together in a cascade style and thus created the global surrogate-based model (GSBM). The level of accuracy was not acceptable as the PRE was extremely large when only 3 surrogate models were deployed. Table 3 shows the performance evaluation of SBVDM, in terms of both accuracy and computational time, when more surrogate models were used. As more surrogate models were applied to the area with big deviations, the PRE of the outputs reduced and SBVDM was able to produce better responses. According to Table 3, the final results of SBVDM with as many as seven surrogate models still could not produce acceptable accuracy.

Using the elaborated flowchart in Figure 2, the final model for SBVDM was constructed using eight different surrogate models. Figure 8 shows the final SBVDM in a cascade module. Since Kriging and ANN could produce better results, these two types of metamodeling techniques were used in the surrogate-based model. Developing the last version of SBVDM required validation in terms of both accuracy and computational time. However, it is important to note that there would always be a trade-off between efficiency and computational cost when using global surrogate models. In this research, it was determined that the computational time

Table 3: Performance evaluation of SBVDM using different number of SMs.

SBVDM	Computational time		Mean PRE (%)				
(number of surrogate models)	(microsec)	X_{dot}	Lateral acceleration	Yaw rate	Bounce	Roll	Pitch
3	109	3.4	89.3	105.4	111.5	93.7	106.5
4	146	3.2	9.0	81.3	123.9	98.4	97.4
5	182	3.2	9.7	6.9	126.3	90.6	101.3
6	216	3.1	9.9	7.2	125.4	89.4	7.3
7	249	3.1	8.8	6.5	9.5	98.4	5.6

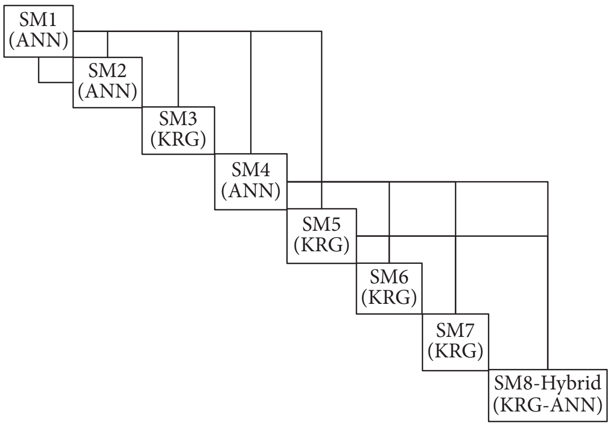


Figure 8: SBVDM developed with 8 surrogate models.

for a single response should be smaller than the time step of 1 millisecond to guarantee a real-time driving simulator.

5.6. Validation of SBVDM. SBVDM was developed to reproduce input-output relationships of the conventional vehicle dynamic model with comparatively lower computational cost. The real-time capability of SBVDM was validated by evaluating the computational delays related to both SBVDM and UTMVDM. The accuracy of SBVDM was evaluated through model validation. The ultimate goal of developing SBVDM was to use it in place of UTMVDM. Thus, it was necessary to evaluate the accuracy of SBVDM in the UTM driving simulator against UTMVDM.

Four different scenarios were used to validate the constructed model, namely, accelerating and hard braking, uphill and downhill, sinusoidal steering maneuver, and vehicle dynamic responses when passing a bump. The model validation of the final version of SBVDM was based on the vehicle dynamic model validation in [25, 37–39]. The results from SBVDM were compared to UTMVDM, which was validated using ADAMS/Car simulation package [36]. If the SBVDM results agreed with that of UTMVDM within certain acceptable error level, the simulation outputs from the surrogate-based model were considered as valid. As was clearly highlighted in [10, 29, 34, 40], error is inevitable when surrogate models are used in a system.

5.6.1. Accelerating and Hard Braking. SBVDM was firstly tested in terms of braking performance. In the said test, the vehicle initially accelerated to 21 m/s, and, subsequently, a hard braking was applied for two and a half seconds. The throttle input was made at zero at the same time. Figure 9 shows the throttle and brake inputs for the test. The brake torque of 300 (Nm) was exerted in the 14th second. Figure 10 illustrates the longitudinal velocity responses of SBVDM compared to that of UTMVDM from the 10th to 20th second. As shown in Figure 9, SBVDM could closely mimic the UTMVDM responses for this scenario.

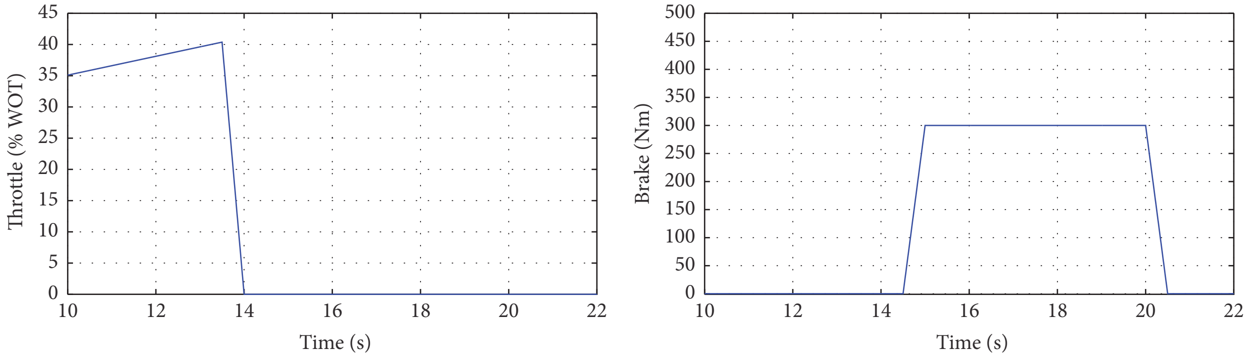
The PRE was less than 1% for the SBVDM responses, which promised the efficiency of SBVDM in producing accurate results for this scenario. The average computational times to calculate a single set of output for UTMVDM and SBVDM for this scenario were 1,274 and 287 microseconds, respectively. As such, it could be concluded that SBVDM was almost 4 times faster than UTMVDM for this scenario.

5.6.2. Uphill and Downhill. In the second test, the simulation was under the same operating condition, which ran on 10 degrees upslope and 10 degrees downslope. The longitudinal velocity of SBVDM was then compared to that of UTMVDM. Figure 11 shows the results of the test from 5th to 10th second of the simulation. The upper lines illustrate the longitudinal velocity responses of SBVDM and UTMVDM for downslope and the bottom lines are related to the upslope condition.

The results indicated that SBVDM could accurately emulate UTMVDM when the road gradient was not zero. The PRE for longitudinal velocity was less than 2%, which promised the efficiency of SBVDM in producing accurate results for running the simulation in similar conditions. The average computational times to calculate a single set of output for UTMVDM and SBVDM for this scenario were 1,234 and 287 microseconds, respectively. Again, this demonstrated that SBVDM was 4 times faster than UTMVDM.

5.6.3. Sinusoidal Steering. As shown in Figure 12, a steering angle similar to the one in [37] was used in this section to validate SBVDM. Figure 13 illustrates the vehicle velocity during the sinusoidal steering. As illustrated in Figures 14–16, SBVDM could accurately produce responses for longitudinal velocity, lateral acceleration, and yaw rate. Comparing the SBVDM outputs with that of UTMVDM, the PRE for these outputs were less than 2%. These results suggested the efficiency of SBVDM in producing accurate responses for similar scenarios.

However, the highest PRE related to vehicle roll angle was approximately 15% in the 8th second of the simulation



(a)

(b)

Figure 9: Throttle and brake torque inputs for accelerating and hard braking scenario.

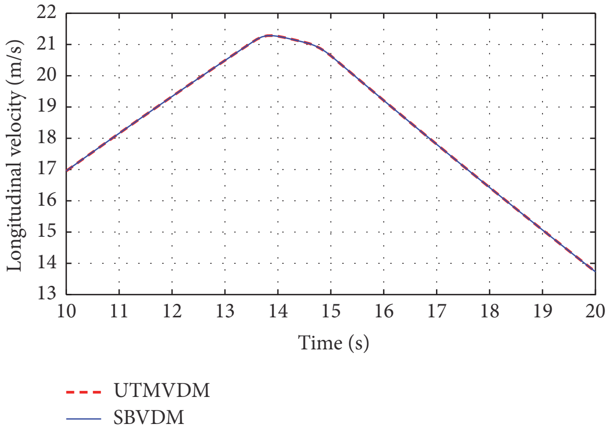


Figure 10: Longitudinal velocity responses for accelerating and hard braking scenario.

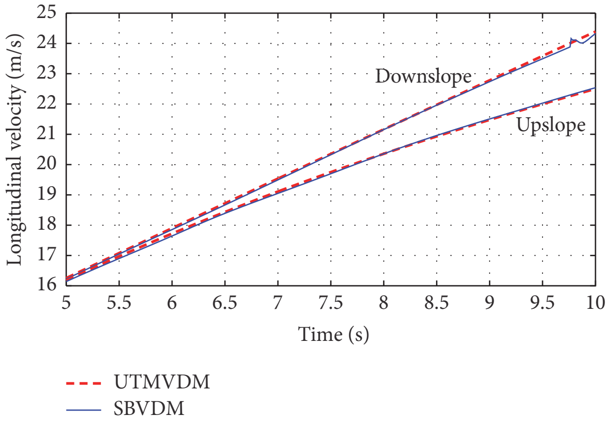


Figure 11: Longitudinal velocity for upslope and downslope road.

(see Figure 17). The discrepancy in the results of the vehicle roll angle may be explained by the fact that the roll angle in UTMVDM was highly dependent on lateral acceleration. Thus, a small deviation in the lateral acceleration output will lead to a big error in the results of roll angle. However, the error for this scenario was acceptable as the mean PRE was 5.609%. In addition, the computational times to calculate a single set of output for UTMVDM and SBVDM for this scenario were 1,351 and 287 microseconds, respectively. The timings indicated that SBVDM was almost 5 times faster than UTMVDM for this scenario.

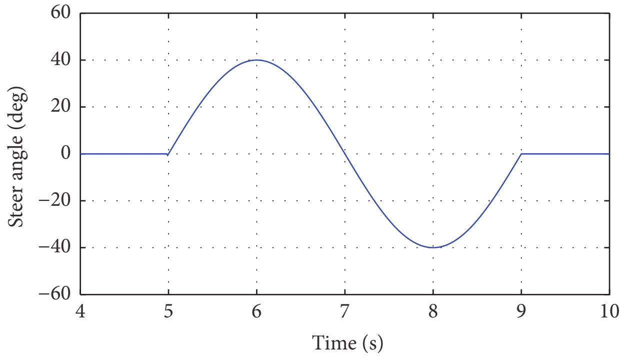


Figure 12: Sinusoidal steer angle used to validate SBVDM.

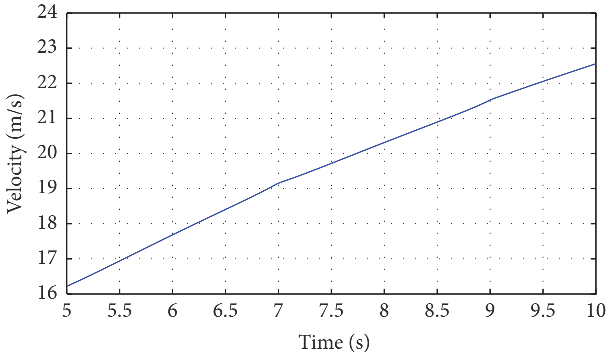


Figure 13: Vehicle velocity during the sinusoidal steering.

5.6.4. Vehicle Dynamics Response When Passing a Bump. In the last testing scenario, the vehicle was made to pass a bump while it was driven on a horizontal flat road. The bump was 0.1 m high and 1 m long. The longitudinal speed for this simulation is as shown in Figure 18. The steering

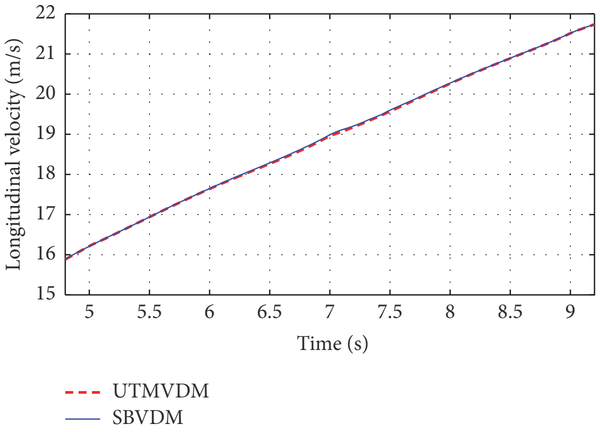


FIGURE 14: Longitudinal velocity results for sinusoidal steering test.

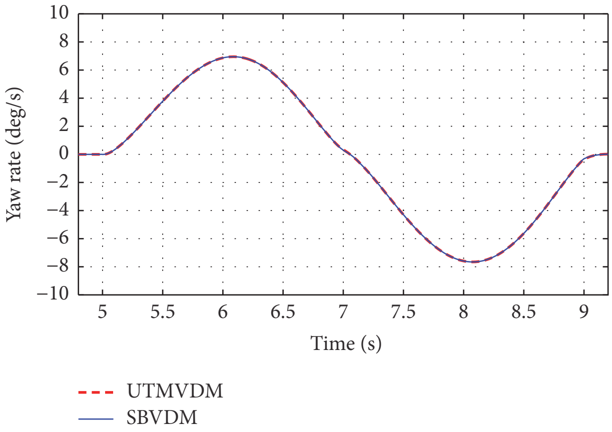


FIGURE 16: Yaw rate for sinusoidal steering test.

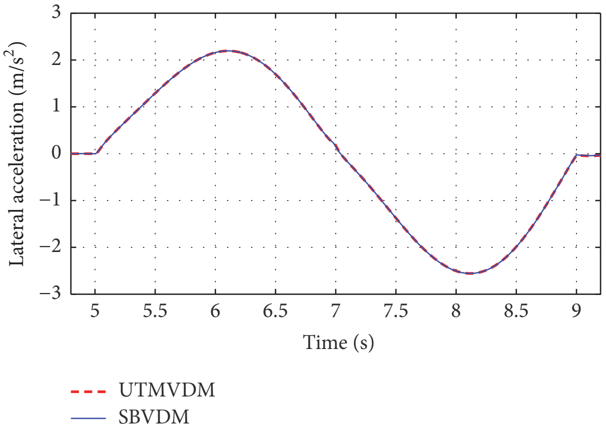


FIGURE 15: Lateral acceleration for sinusoidal steering test.

angle was maintained at zero degree. A simplified full-car suspension model was utilized in UTMVDM, which was able to simulate heave, pitching, and rolling model according to the road inputs. The results of SBVDM were compared to that of UTMVDM to evaluate the performance of the constructed surrogate model.

Figure 19 illustrates the bounce displacement of SBVDM compared to that of UTMVDM. The PRE was below 13%, which made SBVDM reliable for this driving scenario. The large deviations between the UTMVDM and SBVDM results could be seen in the highly nonlinear areas, where it was difficult to construct accurate surrogate models. The vehicle pitch angle results are shown in Figure 20. As can be seen, the PRE was less than 18% for this output. The maximum deviation occurred in the 8th second and PRE was approximately 17.5%. The high error between the results of both SBVDM and UTMVDM was related to the pitch angle and they were mostly seen in highly nonlinear areas. For these areas, constructing more surrogate models can produce a higher level of accuracy. The results indicated that SBVDM could accurately emulate the UTMVDM responses for similar scenarios. The average computational times to calculate a single set of outputs for both UTMVDM and SBVDM for this scenario were 1,373 and 287 microseconds, respectively. Given that, SBVDM was almost 5 times faster than UTMVDM.

5.7. SBVDM Computational Time. The constructed surrogate-based model should be evaluated from the point of view of computational cost as well. The computational times were determined for 1,000 samples during four different runs. Figure 21 clearly shows that SBVDM was much faster than UTMVDM. Indeed, SBVDM was almost five times faster than the conventional model. Considering the lag in data transfer, the time step in SBVDM can be as small as 1,000 microseconds, which guarantees the impression of smooth motion in the virtual driving environment. Thus, if the step size was to shrink to 1,000 microseconds, the computational time in SBVDM would be faster than real time. In contrast, UTMVDM could not compute the vehicle motion in real time for a step size of 1,000 microseconds.

It is also worth noting that the computing time in UTMVDM increased significantly with more complex inputs. This meant that the UTMVDM computational cost changes with the driving scenario. However, the complexity of the inputs did not influence the response time in SBVDM. Thus, the SBVDM computing time was almost the same for the different driving scenarios.

To conclude, the results of this study indicated that SBVDM is an accurate approximation for most of the outputs and is computationally cheaper than UTMVDM. To reiterate, SBVDM was tested using specific vehicle and road inputs, and results suggest that it will be feasible to limit the use of a surrogate model to similar conditions. This will decrease the computational cost of modeling and improves its accuracy. As previously mentioned, there will always be a trade-off between efficiency and computational cost when using global surrogate models. In this case study, the computational time was considerably reduced and an acceptable accuracy was achieved.

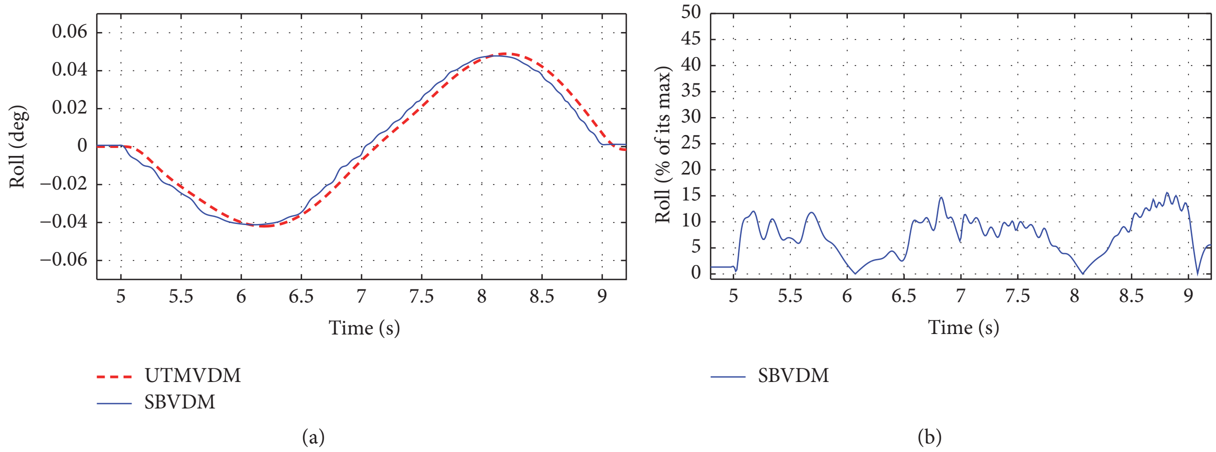


Figure 17: Vehicle roll angle and its PRE for sinusoidal steering test.

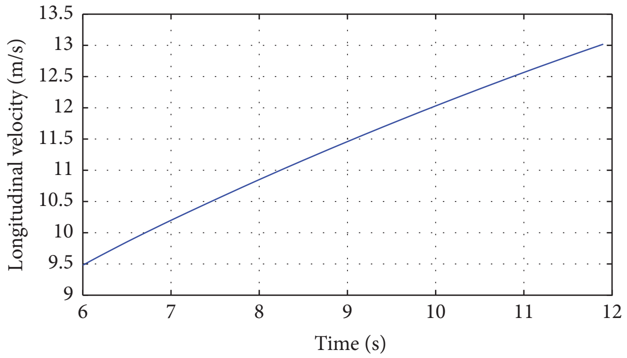


Figure 18: Longitudinal velocity when passing a bump road.

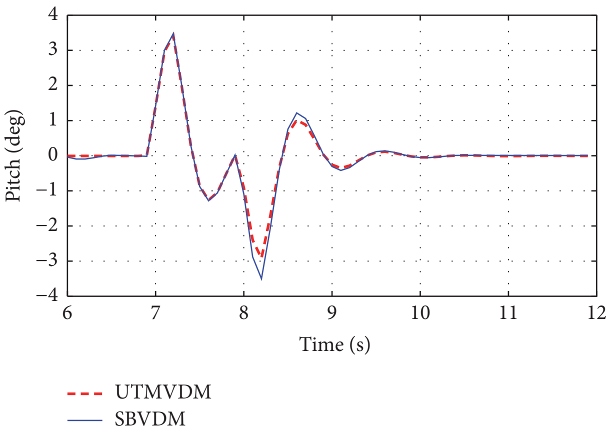


Figure 20: Vehicle pitch angle when passing a bump.

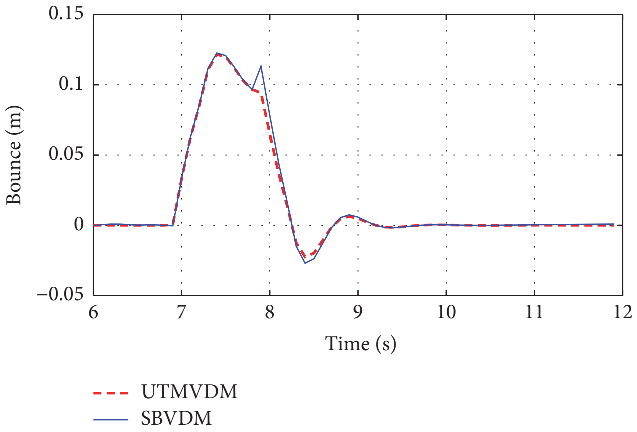


Figure 19: Bounce displacement when passing a bump.

6. Conclusion and Future Works

The development of high-dimensional simulations like vehicle simulators is quite expensive due to their complexity and costly hardware requirements. When a simulation cannot meet its time steps because of a heavy computational burden, it will result in computational delays. In this research, a surrogate-based computational framework was proposed to reduce the computational cost in dealing with multisubsystem simulators. In the proposed framework, decomposition approaches were used to partition the simulation into submodels that could simply be replaced with surrogate models. Well-known metamodeling techniques were employed in the process of creating the surrogate-based simulation. The present research provides a unique methodology to explore the trade-off between accuracy and computational cost by updating the global surrogate-based model with a different number of surrogate models and samples as well as various types of metamodeling techniques.

A case study was presented to assist the understanding of the proposed method. A surrogate-based vehicle dynamic model (SBVDM) was developed step by step using the said framework. The responses of SBVDM were then compared to that of the conventional model. The results showed that the proposed framework could successfully be applied to a vehicle simulator and was also capable of reducing the computational cost in the system. SBVDM could produce acceptable responses and it was five times faster than the conventional model. As such, using SBVDM in a driving simulator allows real-time simulation and increases the fidelity

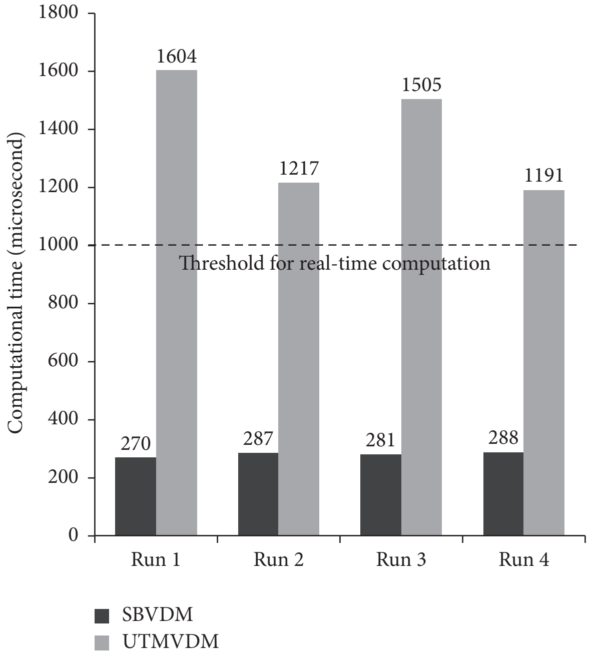


Figure 21: The computational time for responses in SBVDM and conventional VDM.

of the simulation. Thus, the proposed method could reduce the hardware requirements and expenses. Consequently, this will be able to help researchers develop low-cost driving simulations with acceptable fidelity.

Having said that, more research is required to assess the surrogate-based modeling for highly nonlinear responses. As such, future studies should focus on the impact of different sampling strategies and model types on the final results. Studying different aspects of surrogate modeling can help develop a more efficient and accurate surrogate-based simulation for highly nonlinear areas. In addition, further works are needed to estimate the computational limitations and surrogate modeling capabilities to produce responses with different levels of nonlinearity. Another possible area of future research is to investigate new techniques to extend the design space for the global surrogate modeling.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors thank the Malaysian Ministry of Higher Education (MOHE) for their generosity in funding this study.

References

[1] D. J. Fonseca, D. O. Navaresse, and G. P. Moynihan, "Simulation metamodeling through artificial neural networks," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 3, pp. 177–183, 2003.

[2] H. S. Kang, M. K. A. Jalil, and M. Mailah, "A PC-based driving simulator using virtual reality technology," in *Proceedings of the ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry (VRCAI '04)*, pp. 273–277, ACM, Singapore, June 2004.

[3] W.-S. Lee, J.-H. Kim, and J.-H. Cho, "Driving simulator as a virtual reality tool," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 71–76, IEEE, Leuven, Belgium, May 1998.

[4] M. C. Lee, M. K. Park, W. S. Yoo, K. Son, and M. C. Han, "Sliding mode control of real-time PNU vehicle driving simulator and its performance eveluation," *JSME International Journal, Series C*, vol. 46, no. 2, pp. 557–564, 2003.

[5] R. W. Allen, T. J. Rosenthal, and Z. Parseghian, "Low cost driving simulation for research, training and screening applications," *SAE Technical Papers*, Article ID 950171, 12 pages, 1995.

[6] M. Karkee, B. L. Steward, A. G. Kelkar, and Z. T. Kemp II, "Modeling and real-time simulation architectures for virtual prototyping of off-road vehicles," *Virtual Reality*, vol. 15, no. 1, pp. 83–96, 2011.

[7] M. Krebs, "Vehicle modeling for high-dynamic driving simulator applications," in *Proceedings of the 1st Human-Centered Transportation Simulation Conference*, Iowa City, Iowa, USA, 2001.

[8] W.-S. Lee, "Compensation of transport delay in driving simulator," in *Proceedings of the American Control Conference*, pp. 1050–1055, June 1991.

[9] E. J. Haug, D. Negrut, R. Serban, and D. Solis, "Numerical methods for high-speed vehicle dynamic simulation," *Mechanics of Structures and Machines*, vol. 27, no. 4, pp. 507–533, 1999.

[10] A. Ghazizadeh, A. Fahim, and M. El-Gindy, "Neural networks representation of a vehicle model: 'Neuro-Vehicle (NV)'," *International Journal of Vehicle Design*, vol. 17, no. 1, pp. 55–75, 1996.

[11] H. Holzmann, O. Nelles, C. Halfmann, and R. Isermann, "Vehicle dynamics simulation based on hybrid modeling," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '99)*, pp. 1014–1019, September 1999.

[12] V. Karri and D. Butler, "Using artificial neural networks to predict vehicle acceleration and yaw angles," in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP '02)*, pp. 1915–1919, IEEE, Singapore, November 2002.

[13] L. Qiang, W. Huiyi, and G. Konghui, "Identification and control of four-wheel-steering vehicles based on neural network," in *Proceedings of the IEEE International Vehicle Electronics Conference (IVEC '99)*, pp. 250–253, Changchun, China, September 1999.

[14] S. J. Rutherford and D. J. Cole, "Modelling nonlinear vehicle dynamics with neural networks," *International Journal of Vehicle Design*, vol. 53, no. 4, pp. 260–287, 2010.

[15] Y. U. Yim and S.-Y. Oh, "Modeling of vehicle dynamics from real vehicle measurements using a neural network with two-stage hybrid learning for accurate long-term prediction," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1076–1084, 2004.

[16] C. L. Bloebaum, "Coupling strength-based system reduction for complex engineering design," *Structural Optimization*, vol. 10, no. 2, pp. 113–121, 1995.

[17] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies for high-dimensional design problems," in *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and*

Optimization Conference (MAO '08), British Columbia, Canada, September 2008.

[18] K. F. Hulme, *The Design of a Simulation-Based Framework for the Development of Solution Approaches in Multidisciplinary Design Optimization*, Faculty of the Graduate School, University of New York, Buffalo, NY, USA, 2000.

[19] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, "Ensemble of surrogates," *Structural and Multidisciplinary Optimization*, vol. 33, no. 3, pp. 199–216, 2007.

[20] T. W. Simpson, D. K. Lin, and W. Chen, "Sampling strategies for computer experiments: design and analysis," *International Journal of Reliability and Applications*, vol. 2, no. 3, pp. 209–240, 2001.

[21] D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, 2008.

[22] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen, "Metamodels for computer-based engineering design: survey and recommendations," *Engineering with Computers*, vol. 17, no. 2, pp. 129–150, 2001.

[23] Y. S. Chiew, *Design of a Six Degree of Freedom Motion Platform for Vehicle Driving Simulator Application*, Universiti Teknologi Malaysia, Faculty of Mechanical Engineering, Johor Bahru, Malaysia, 2009.

[24] N. Fouladinejad, N. Fouladinejad, M. K. Abd Jalil, and J. M. Taib, "Modeling virtual driving environment for a driving simulator," in *Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE '11)*, pp. 27–32, IEEE, Penang, Malaysia, November 2011.

[25] H. S. Kang, *Design and development of a virtual reality fixed base driving simulator [M.S. thesis]*, Department of Mechanical Engineering, Universiti Teknologi Malaysia, 2008.

[26] N. Fouladinejad, N. Fouladinejad, M. K. Abdul Jalil, and J. Mohd Taib, "Development of a surrogate-based vehicle dynamic model to reduce computational delays in a driving simulator," *SIMULATION*, vol. 92, no. 12, pp. 1087–1102, 2016.

[27] D. V. Steward, "Design structure system: a method for managing the design of complex systems," *IEEE Transactions on Engineering Management*, vol. EM-28, no. 3, pp. 71–74, 1981.

[28] J. Sobieszczanski-Sobieski, J. S. Agte, and R. R. Sandusky Jr., "Bilevel integrated system synthesis," *AIAA Journal*, vol. 38, no. 1, pp. 164–172, 2000.

[29] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design," *Journal of Machine Learning Research*, vol. 11, pp. 2051–2055, 2010.

[30] J.-F. M. Barthelemy and R. T. Haftka, "Approximation concepts for optimum structural design—a review," *Structural Optimization*, vol. 5, no. 3, pp. 129–144, 1993.

[31] W. C. M. Van Beers and J. P. C. Kleijnen, "Kriging for interpolation in random simulation," *Journal of the Operational Research Society*, vol. 54, no. 3, pp. 255–262, 2003.

[32] P. Hajela and L. Berke, "Neural networks in structural analysis and design: an overview," *Computing Systems in Engineering*, vol. 3, no. 1–4, pp. 525–538, 1992.

[33] B. Cheng and D. M. Titterington, "Neural networks: a review from a statistical perspective," *Statistical Science*, vol. 9, no. 1, pp. 2–30, 1994.

[34] M. F. Hussain, R. R. Barton, and S. B. Joshi, "Metamodeling: radial basis functions, versus polynomials," *European Journal of Operational Research*, vol. 138, no. 1, pp. 142–154, 2002.

[35] P. Guarneri, G. Rocca, and M. Gobbi, "A neural-network-based model for the dynamic simulation of the tire/ suspension system while traversing road irregularities," *IEEE Transactions on Neural Networks*, vol. 19, no. 9, pp. 1549–1563, 2008.

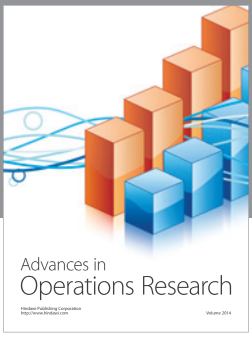
[36] R. L. Mueller, *Full Vehicle Dynamics Model of a Formula SAE Racecar Using ADAMS/Car*, Texas A&M University, College Station, Tex, USA, 2005.

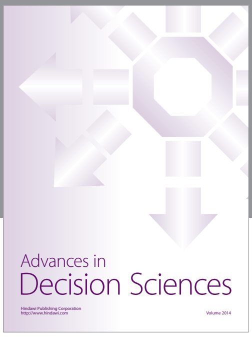
[37] G. P. Bertollini, C. M. Johnston, J. W. Kuiper, J. C. Kukula, M. A. Kulczycka, and W. E. Thomas, "The general motors driving simulator," *SAE Technical Papers*, 1994.

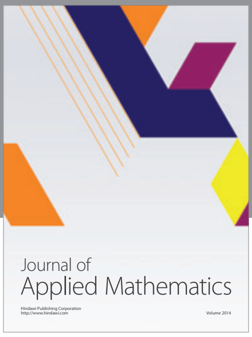
[38] G. J. Heydinger, C. Schwarz, M. Salaani, and P. Grygier, "Model validation of the 2006 BMW 330i for the National Advanced Driving Simulator," SAE Technical Paper 2007-01-0817, 2007.

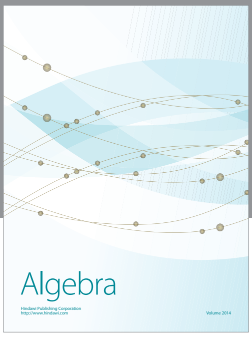
[39] R. Singh and M. K. Hada, "Vehicle dynamics modelling and simulation of a four wheeled ground vehicle for a pc based driving simulator," *SAE Technical Papers*, 2007.

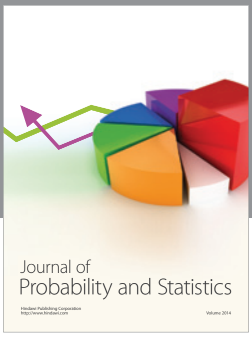
[40] D. Gorissen, I. Couckuyt, E. Laermans, and T. Dhaene, "Multi-objective global surrogate modeling, dealing with the 5-percent problem," *Engineering with Computers*, vol. 26, no. 1, pp. 81–98, 2010.


Advances in
Operations Research

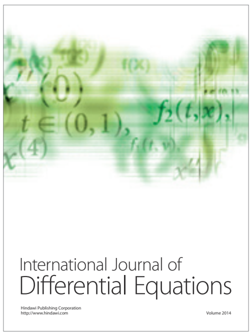

Advances in
Decision Sciences


Journal of
Applied Mathematics

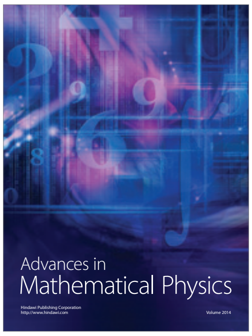

Algebra

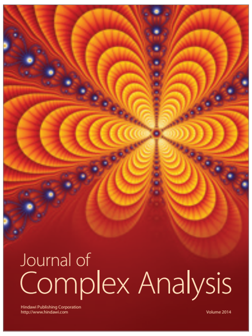

Journal of
Probability and Statistics

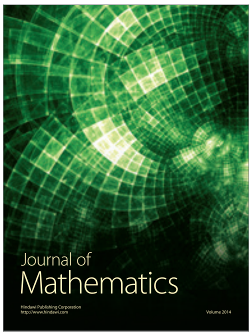
The Scientific
World Journal

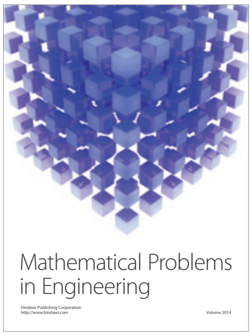

International Journal of
Differential Equations

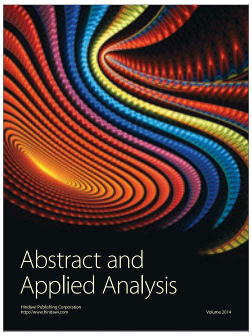
International Journal of
Combinatorics

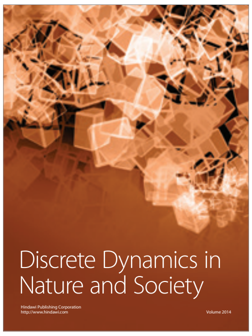

Advances in
Mathematical Physics

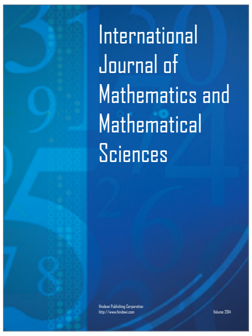

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

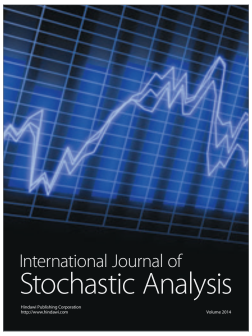

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization