Optimising Control and Prediction Horizons of a Model Predictive Control-Based Motion Cueing Algorithm Using Butterfly Optimization Algorithm

Mohammad Reza Chalak Qazani
the Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
m.r.chalakqazani@gmail.com

Seyed Mohammad Jafar Jalali
the Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
mohammadjj.it@gmail.com

Houshyar Asadi
the Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
houshyar.asadi@deakin.edu.au

Saeid Nahavandi
the Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
saeid.nahavandi@deakin.edu.au

***Abstract*— The Motion Cueing Algorithm (MCA) oversees regenerating the motion feeling of the real vehicle for the simulation-based motion platform (SBMP) within the physical limitations. Model Predictive Control (MPC) is recently employed as an MCA, which is called MPC-based MCA due to the consideration of the plant's boundaries in finding the optimal input signal. The computational load of the MPC directly relates to the control horizon and prediction horizon of the MPC. In this paper, a new optimisation method using butterfly optimisation algorithm is developed to find the optimal control horizon and prediction horizon of MPC-based MCA. The proposed method reduces the time of the tuning process of the MPC-based MCA, which is usually carried out via trial-and-error and genetic algorithm methods. Also, the trial-and-error method increases the motion sensation error and insufficient usage of the SBMP. The model is validated using MATLAB simulation environment, and the outcomes show that the developed butterfly optimisation algorithm will lead better motion sensation with less wrong motion signals and low computational burden compared with the trial-and-error and genetic algorithm method.**

Keywords*— *motion cueing algorithm, model predictive control, MPC horizons, meta-heuristic algorithm, butterfly optimisation algorithm, genetic algorithm

I. Introduction

The simulation-based motion platforms (SBMPs) are capable of regenerating the 6-degree-of-freedom (6-DoF) motions. They have been employed recently in research laboratories such as medical or virtual reality because of their benefits such as reduction of cost, time and damage [1-3]. The SBMPs can be employed for increasing testing the newly designed vehicle, road safety, training the drivers and study the motion sickness [4]. The parallel and serial manipulators are typically used as SBMP [5-7]. As the parallel and serial manipulators are restricted due to the limitations of their joints [8-10], it is not possible to implement the actual motion signals from the real vehicle to the SBMPs [11-13].

A. *Motion Cueing Algorithm*

It is possible to regenerate the exact motion sensation of the real car using motion cueing algorithm (MCA) with respecting the SBMP limitations [14]. The main reason for the motion sickness is due to the SBMP limitations [4]. Then, it changes the perceived motion signals via the vestibular system compared with the visually perceived motion signals which causes motion sickness.

The vestibular system is a susceptible sensor that is in charge of sensing motions and keeping the postural stability. The vestibular system is composed of the semicircular system and otolith organs for sensing rotational and translational motions, respectively [15, 16]. The otolith organs cannot differentiate between the sustained acceleration and tilt. Therefore, the tilt coordination channel can be used through MCA to create a sustainable acceleration sensation via somatogravic illusion [17]. Also, it is essential to respect the human rotational perception threshold via the semicircular system to avoid the sensation of angular velocity [15].

The first MCA is introduced by Conrad and Schmidt [18] known as the classical washout filter. The simplicity, easy tuning, low computational burden and accessibility are the advantages of the classical washout filter. However, there are some drawbacks such as constant parameters, worst-case motion scenario tuning method and neglection of the human vestibular model. Schmidt [19] introduced the adaptive washout filter to fluctuate the constant indexes of the classical washout filter. Asadi et al. [20] employed the fuzzy logic technique in developing the adaptive washout filter, which can consider the SBMP limitations and the motion sensation error for the SBMP user to generate the accurate motion signals. The adaptive washout filter suffers from instability, oscillatory response, high computational burden and undesirable spikes. The optimal form of the classical washout filter with cogitation of the human vestibular model is called optimal washout filter [21]. Unlikely, the optimal washout filter cannot consider the SBMP limitations, which is the most critical drawback of the mentioned MCA.

Dagdelen et al. [22] introduced the idea of the optimal MCA with the determination of the physical limitations known as model predictive control-based (MPC-based) MCA. The MPC recalculate the optimal input signal to minimise the error of the plant's outputs in the prediction horizon based on the dynamical model of the plant and its' constraints. The explicit MPC-based MCA by the invariant set method is introduced in [23] to reach the fast response for the longitudinal channel with 2-DoF motion signals. Bruschetta et al. [24] introduced the first nonlinear MPC-based MCA using ACADO toolkit [25]. Qazani et al. [26] developed the MPC-based MCA for a midsize SBMP using the decoupled model of the human vestibular system to control the tilt angle more precisely. Recently, Qazani et al. [27, 28] employed the time-varying MPC in generation of the high accurate MCA with consideration of active joints limitations.

B. Tuning Problem of MCA

The tuning of the MCA indexes is challenging from the introduction of this technique. One of the methods to tune the parameters is trial-and-error method. The tuning parameters of the MPC-based MCA are complicated as MPC has a lot of adaptable indexes such as weights and horizons. Asadi et al. [29] employed a genetic algorithm in developing the human perception-based MCA. Mohammadi et al. [30] introduced the method of the multi-objective GA with the interaction of the human case to optimise the tunning weights of the MPC-based MCA. The computational load of the MPC directly depends on the MPC horizons. MPC horizons determine the size of the QP problem. It should be chosen wisely to reach the relevant results and acceptable computational burden. Mohammadi et al. [31] employed the GA to find the optimal MPC horizons. Unfortunately, as GA is a random selection of individuals, the convergence speed is slow for finding accurate solutions. Accordingly, in tuning the MCA parameters we still need to develop accurate algorithms to mitigate local minimum problems. This paper aims to apply a better global optimisation algorithm called butterfly optimisation algorithm (BOA). The inspiration for this meta-heuristic algorithm is the hunting actions of butterflies in universe. Through several challenging and complex benchmarking problems the BOA demonstrated its supremacy to escape from local optima. In this paper, two meta-heuristic algorithms, including GA and BOA, are employed to find the optimal MPC horizons. Each strategy is applied ten times to show the accuracy and efficiency of the algorithms.

Section II explains the central concept of the MPC algorithm for the MCA. Section III presents the meta-heuristic algorithms and the objective function of MCA. Section IV demonstrates the results and discussion of BOA tuning method and are compared with GA and trial-and-error tuning methods. The conclusions and remarks are represented in Section V.

II. MPC-BASED MCA

The sensed specific force via the real vehicle and SBMP driver along three axis based on the tilt-coordination channel can be calculated as:

$$\mathbf{S}_{\mathrm{F}} = \begin{cases} f_x \triangleq a_x + g\theta \\ f_y \triangleq a_y - g\phi \\ f_z \triangleq a_z - g \end{cases} \tag{1}$$

where a_x, a_y and a_z are linear acceleration along *x*-, *y*- and *z*-axis. Also, ϕ and θ are *roll*- and *pitch*-angle.

The technique known as receding horizon [32] is employed to apply the first argument of the extracted input signal to the mechanism. The MPC is designed based on the discretised model of the plant as follows:

$$\mathbf{x}_{\mathrm{m}}(t+1) = \mathbf{A}_{\mathrm{m}}\mathbf{x}_{\mathrm{m}}(t) + \mathbf{B}_{\mathrm{m}}\mathbf{u}_{\mathrm{m}}(t) \tag{2.a}$$

$$\mathbf{y}(t) = \mathbf{C}_{\mathrm{m}}\mathbf{x}_{\mathrm{m}}(t) \tag{2.b}$$

where $\mathbf{A}_{\mathrm{m}}$, $\mathbf{B}_{\mathrm{m}}$ and $\mathbf{C}_{\mathrm{m}}$ are matrices of MPC state space model. Also, $\mathbf{u}_{\mathrm{m}}$ is the input motion signal, including linear acceleration and rotational velocity signal.

In order to find the translational displacement, translational velocity and rotational displacement of the SBMP from the inputs of the system, the integral model is formulated as:

$$\dot{\mathbf{x}}_{\mathrm{d}} = \mathbf{A}_{\mathrm{d}}\mathbf{x}_{\mathrm{d}} + \mathbf{B}_{\mathrm{d}}\mathbf{u}_{\mathrm{s}} \tag{3}$$

where $\mathbf{x}_{\mathrm{d}}$, $\mathbf{A}_{\mathrm{d}}$ and $\mathbf{B}_{\mathrm{d}}$ are:

$$\dot{\mathbf{x}}_{\mathrm{d}} = \left[\iint \ddot{X} dt^2 \quad \int \ddot{X} dt \quad \theta\right] \tag{4.a}$$

$$\mathbf{A}_{\mathrm{d}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.b}$$

$$\mathbf{B}_{\mathrm{d}} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{4.c}$$

There are limitations on the input rate, input and output in MPC-based MCA, then $\Delta\mathbf{u}(t) \triangleq \mathbf{u}(t) - \mathbf{u}(t-1)$ and $\Delta\mathbf{x}_{\mathrm{m}}(t) \triangleq \mathbf{x}_{\mathrm{m}}(t) - \mathbf{x}_{\mathrm{m}}(t-1)$ are defined as a variety of input and states to consider the limitations of input rates. The augment model can be demonstrated as:

$$\mathbf{x}(t+1) = \begin{bmatrix} \mathbf{A}_{\mathrm{m}} & \mathbf{0} \\ \mathbf{C}_{\mathrm{m}}\mathbf{A}_{\mathrm{m}} & \mathbf{I} \end{bmatrix}\mathbf{x}(t) + \begin{bmatrix} \mathbf{B}_{\mathrm{m}} \\ \mathbf{C}_{\mathrm{m}}\mathbf{B}_{\mathrm{m}} \end{bmatrix}\Delta\boldsymbol{u}(t) \tag{5.a}$$

$$\mathbf{y}(t) = [\mathbf{0} \quad \mathbf{I}]\mathbf{x}(t) \tag{5.b}$$

where $\Delta\mathbf{u}$ and $\mathbf{x}(t)$ are the control input state vector of the augment state space model. Eq. (5) is rewritten as:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\Delta\mathbf{u}(t) \tag{6.a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{6.b}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are the matrices of the augment state space model of MPC.

The model predictive control aims to calculate the optimal input signal along the control horizon N_C that reaches the output signal to the actual signal along the prediction horizon N_P. The prediction of the states and outputs in the n sample time can be calculated as:

$$\mathbf{x}(t+1|t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\Delta\mathbf{u}(t) \tag{7.a}$$

$$\mathbf{y}(t+1|t) = \mathbf{C}\mathbf{A}\mathbf{x}(t) + \mathbf{C}\mathbf{B}\Delta\mathbf{u}(t) \tag{7.b}$$

$$\cdots \tag{7.c}$$

$$\mathbf{x}(t+N_P|t) = \mathbf{A}^{N_P}\mathbf{x}(t) + \mathbf{A}^{N_P-1}\mathbf{B}\Delta\mathbf{u}(t) + \mathbf{A}^{N_P-2}\mathbf{B}\Delta\mathbf{u}(t+1) + \cdots + \mathbf{A}^{N_P-N_c}\mathbf{B}\Delta\mathbf{u}(t+N_c-1) \tag{7.d}$$

$$\mathbf{y}(t+N_P|t) = \mathbf{CA}^{N_P}\mathbf{x}(t) + \mathbf{CA}^{N_P-1}\mathbf{B}\Delta\mathbf{u}(t) + \mathbf{CA}^{N_P-2}\mathbf{B}\Delta\mathbf{u}(t+1) + \cdots + \mathbf{CA}^{N_P-N_C}\mathbf{B}\Delta\mathbf{u}(t+N_c-1) \quad (7.e)$$

where T_s and $\mathbf{x}(t+n|t)$ are the sample time of the control and the state in the n sample time, respectively.

The outputs only relate to the input succession and the present state. The new vector form of inputs and outputs of Eq. (7) called $\Delta\mathbf{U}$ and $\mathbf{Y}$ where $\Delta\mathbf{U} \in \mathbb{R}^{(N_C \cdot N_{in})\times 1}$ and $\mathbf{Y} \in \mathbb{R}^{(N_P \cdot N_{out})\times 1}$. N_{in} and N_{out} are the size of $\Delta\mathbf{u}$ and $\mathbf{Y}$, respectively. Then, the input-output evaluation of the model can be presented as follows:

$$\mathbf{Y} = \mathbf{f} + \mathbf{G}\Delta\mathbf{U} \quad (8)$$

where $\mathbf{f}$ and $\mathbf{G}$ are matrices as follows:

$$\mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{N_P} \end{bmatrix} \mathbf{x}(t) \quad (9.a)$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N_P-1}\mathbf{B} & \mathbf{CA}^{N_P-2}\mathbf{B} & \cdots & \mathbf{CA}^{N_P-N_C}\mathbf{B} \end{bmatrix} \quad (9.b)$$

If $\mathbf{R_s}$ is the reference signal known as sensed specific force and angular velocity of the real vehicle, the cost function is determined as:

$$\mathbf{J}(\Delta\boldsymbol{U}) = \left(\mathbf{R}_{Ref} - \mathbf{Y}\right)^{\mathrm{T}}\mathbf{Q}\left(\mathbf{R}_{Ref} - \mathbf{Y}\right) + \mathbf{U}^{\mathrm{T}}\mathbf{S}\mathbf{U} + \Delta\mathbf{U}^{\mathrm{T}}\mathbf{R}\Delta\mathbf{U} \quad (10)$$

where $\mathbf{S}$, $\mathbf{R}$ and $\mathbf{Q}$ are the block diagonal weighting parameters associated with the input and input rate and output, respectively. Also, $\mathbf{Y}$, $\mathbf{U}$ and $\Delta\mathbf{U}$ are the predicted future output vector, input succession and control action increment series, respectively. The input succession $\mathbf{U}$ can be determined as:

$$\mathbf{U} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \end{bmatrix}}_{\mathbf{T}} \Delta\mathbf{U} + \underbrace{\begin{bmatrix} \mathbf{u}(t-1) \\ \mathbf{u}(t-1) \\ \vdots \\ \mathbf{u}(t-1) \end{bmatrix}}_{\bar{\mathbf{U}}_\mathbf{i}} \quad (11)$$

The new cost function can be found by substituting the Eq. (8) and Eq. (11) in Eq. (10) as follows:

$$\mathbf{J}(\Delta\mathbf{U}) = \left(\mathbf{R}_{Ref} - \mathbf{f} - \mathbf{G}\Delta\mathbf{U}\right)^{\mathrm{T}}\mathbf{Q}\left(\mathbf{R}_{Ref} - \mathbf{f} - \mathbf{G}\Delta\mathbf{U}\right) + (\mathbf{T}\Delta\mathbf{U} + \bar{\mathbf{U}}_\mathbf{i})^{\mathrm{T}}\mathbf{S}(\mathbf{T}\Delta\mathbf{U} + \bar{\mathbf{U}}_\mathbf{i}) + \Delta\mathbf{U}^{\mathrm{T}}\mathbf{R}\Delta\mathbf{U} \quad (12)$$

where $\mathbf{R}$, $\mathbf{S}$ and $\mathbf{Q}$ are the block diagonal weighting matrices for penalising the errors of the input rate, input and output, respectively.

$\mathbf{J}(\Delta\mathbf{U})$ is minimised over $\mathbf{U}$ and $\Delta\mathbf{U}$. With a definition of the $\mathbf{H} \triangleq 2(\mathbf{G}^{\mathrm{T}}\mathbf{Q}\mathbf{G} + \mathbf{R} + \mathbf{T}^{\mathrm{T}}\mathbf{S}\mathbf{T})$ and $\mathbf{g} = 2(\mathbf{G}^{\mathrm{T}}\mathbf{Q}(\mathbf{f} - \mathbf{R}_{\mathrm{Ref}}) + \mathbf{T}^{\mathrm{T}}\mathbf{S}\bar{\mathbf{U}}_i)$, the Eq. (13) can be extracted as:

$$\mathbf{J}(\Delta\mathbf{U}) = \frac{1}{2}\Delta\mathbf{U}^{\mathrm{T}}\mathbf{H}\Delta\mathbf{U} + \Delta\mathbf{U}^{\mathrm{T}}\mathbf{g} \quad (13)$$

The combination of Eq. (13) with linear constraints is the classical expression of the QP problem that is used to optimise the $\Delta\mathbf{U}$. Then, the constraints of the MPC model should be formulated based on $\mathbf{Ax} < \mathbf{b}$ as follows:

$$\begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{bmatrix} \Delta\mathbf{U} \leq \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{n}_3 \end{bmatrix} \quad (14)$$

The constraints are interpreted into three groups, such as the inputs rates' limitations, inputs' limitations and outputs' limitations. $\mathbf{M}_1$ and $\mathbf{n}_1$ matrices relate to the limitations of input rates. $\mathbf{M}_2$ and $\mathbf{n}_2$ are defined as a constraint's matrices of the input. $\mathbf{M}_3$ and $\mathbf{n}_3$ connect the output results, as shown in Table I.

The vestibular system is in charge of the sensation of the angular and linear motions. Asadi et al. [15, 16] discovered the most reliable transfer function model of the human otolith organs and the semicircular system as follow:

$$\frac{\hat{f}(s)}{f(s)} = K_{OTO}\left(\frac{(\tau_a s+1)}{(\tau_L s+1)(\tau_s s+1)}\right) \quad (15.a)$$

$$\frac{\widehat{\omega}}{\omega} = \left(\frac{\tau_1 \tau_a s^2}{(1+\tau_a s)(1+\tau_1 s)}\right) \quad (15.b)$$

where $\hat{\mathrm{f}}$ and $\widehat{\omega}$ are the sensed specific force and sensed angular velocity. Also, K_{OTO}=0.4, τ_L=5.3 (s), τ_s=0.016 (s), τ_a=13.2 (s), τ_1=5.3 (s), and τ_a=30 (s) in the longitudinal channel.

III. Meta-heuristic Algorithm

Meta-heuristic optimisation algorithms are widely used in engineering applications because of easy implementation, avoiding local optima and popularity. GA [33] as an evolutionary-based algorithm is introduced by inspiration of the natural evolution. Mohammadi et al. [34] introduced the open access GA library using C++ to be used in MCA. Arora and Singh [35] introduced BOA recently that inspires via the behaviour of butterfly for foraging and mating. Both meta-heuristic algorithms are employed in this study to minimise the motion sensation error and computational load of the model.

A. Cost Function

The essential factors of the MPC-based MCA are weights and control horizons. The control horizon and prediction horizon affect the size of the QP problem, which is the main reason for the computational load of the MPC algorithm. Prediction horizon enhances the size of the matrix in the QP problem. The higher prediction horizon improves the results of the MPC as it increases the size of matrices for longer forecasting the system. However, higher prediction horizon increases the computational load of the system, and it should be selected wisely between the best results while having an acceptable computational burden. Control horizon decides the degree of freedom for control input. Higher control horizon increases the dexterity of the system to react quickly for the variations. Still, the upper control horizon can cause adverse effects such as fluctuation and high computational burden. In this paper, an off-line optimisation algorithm is utilised to find the optimal MPC horizons in order to minimise the cost function as follows:

$$J(N_P, N_c) = J_{MCA} + w_{com}J_{com} \quad (16)$$

where J_{MCA} and J_{com} are the cost functions related to the MCA and computational burden of the system. w_{com} is the computational burden cost function. The cost function J_{MCA} is defined as follows:

$$J_{MCA} = w_{\hat{f}_x}\int(\hat{f}_x - f_x)dt + w_{\widehat{\omega}_\theta}\int(\widehat{\omega}_\theta - \omega_\theta)dt + w_x\int x^2 dt + w_{V_x}\int {V_x}^2 dt + w_\theta\int \theta^2 dt + w_{\omega_\theta}\int {\omega_\theta}^2 dt +$$

TABLE I THE WORKIN AREA LIMITATION OF THE HEXAPOD SBMP IN LONGITUDINAL CHANNEL

Index	Position	Velocity	Acceleration
x-axis	±0.15 m	±15 m/s	±35 m/s²
Pitch-angle	±0.1047 rad	±2.35 rad /s	±43.63 rad /s²

$$w_{a_x}\int a_x{}^2dt + w_{\alpha_\theta}\int \alpha_\theta{}^2dt + w_{Jerk_x}\int Jerk_x{}^2dt \qquad (17)$$

where $w_{\hat{f}_x}$, $w_{\widehat{\omega}_\theta}$, w_x, w_{V_x}, w_θ, w_{ω_θ}, w_{a_x}, w_{α_θ} and w_{Jerk_x} are the influence factors of the sensed specific force, sensed angular velocity, translational displacement, translational velocity, rotational displacement, rotational velocity, translational acceleration, rotational acceleration and translational jerk of the MPC-based MCA. The trial-and-error can determine them for the cost function.

The influence of the MPC horizons on the computational load of the system is determined using a cost function J_{com}. It should be a logical function that relates the control horizon and prediction horizon to the computational time of the MPC. [31] investigated the logical cost function formulation for the MPC-based MCA via below conditions.

$N_c \le N_P$: The prediction horizon should be greater or equal to the control horizon.

$J_{com} \le Max$: It is defined to avoid the results with high simulation time.

Then, the computational cost function using the function fitting is obtained as [31]:

$$J_{com} = 0.03N_P + 0.039N_c{}^2 + 0.00054N_P{}^2 + 0.0028N_PN_c \qquad (18)$$

B. Genetic Algorithm

The GA is utilized to solve the constrained and unconstrained problems as an evolutionary-based algorithm. GA is accessible in engineering applications such as optimising MCAs [34]. GA was introduced by Holland [33] based on the theory of Darwinism. GA is a random selection of individuals from the present population. Then, GA extracts the parents and employs them to generate the next generation's children.

The GA adjustment parameters for finding the optimal horizons of the MPC-based MCA is shown in Table II, including population size, maximum generation, crossover and function tolerance. These parameters affect the precision and convergence speed of the GA. The result of the GA can be guaranteed using a suitable crossover and mutation rates [36]. In this research, the low mutation rate and a high crossover fraction are selected based on the recommended settings by [33]. The higher population size increases the computational load of the GA without noticeable improvement while the lower population size reduces the efficiency of the GA. Then, the population size of GA for extracting the optimal horizons of the MPC-based MCA is chosen to be 100.

C. Butterfly Optimization Algorithm

BOA has been first proposed by Arora and Singh [35], and its idea is based on the butterfly's behaviour in nature. In contrast to well-known metaheuristics such as GA which keeps

TABLE II. GA COEFFICIENT PARAMETERS

Parameters	Value
Variables	$[N_P; N_c]$
Population Size	100
Maximum Generation	45
Upper Limit	[2000; 200]
Lower Limit	[5; 2]
Crossover Rate	0.8
Mutation Function	Adaptive Feasible
Function Tolerance	10^{-6}

the qualified solutions, BOA does not get rid of any solutions in the search space which means each solution has an equal chance for improvement with the new solution. Another significant difference of BOA compared with traditional evolutionary algorithms which compute the value of fitness function for generating the new solutions. The butterflies considered as search agents in BOA, move randomly towards each other's and create more new solutions.

BOA follows three general stages, including Initialization, Iteration and Final Stages. In the Initialization Stage, the algorithm determines the cost function, solution space and the boundary variables of the problem by creating a population of individuals. For the second stage, BOA searches the position of butterflies in an iteration procedure randomly by calculating and storing the fitness values and finally, in the last step, BOA reaches to the termination in which it finds the best solution for the optimisation problem.

From the mathematical perspective, the butterflies acting as the search agents for the BOA and creates fragrance at their positions using $k = cI^a$ where c, k, a and I is the sensory modality, the perceived magnitude of the fragrance, the power exponent dependent on modality and the stimulus intensity, respectively. This algorithm is based on two critical steps, including universe search stage and local search stage.

From universe search stage perspective, the butterflies take a step in the direction of the fittest butterfly for creating the solution h^* shown in Eq. (19) as:

$$z_i^{t+1} = z_i^t + (r^2h^* - z_i^t)k_i \qquad (19)$$

where the solution vector z_i for ith butterfly in t iteration, is represented by z_i^t. h^* is for the current best solution found between each generated solution in the current iteration.

r represents a random number in the interval of [0, 1] and k_i is devoted to the Fragrance of ith butterfly in the search space.

Eq. (20) Shows the procedure of local search stage as follows:

$$z_i^{t+1} = z_i^t + \left(r^2z_j^t - z_k^t\right)k_i \qquad (20)$$

where z_j^t and z_k^t represent j^{th} and k^{th} butterflies from the solution space, respectively. It should be noted that a local random walk is generated by Eq. (20), If both z_j^t and z_k^t belong to the same solution.

Butterflies look for activities such as food and partner using strategies described in local and universe stages. Due to the proximity of the physical and various other factors such as rain and windy conditions, the needs related to food and partner have a notable fraction probability of p in such activities. In this algorithm, this probability looks for common strategies between universe and local searches in the search space of the problem. By meeting the stopping criteria, BOA finds the best fitness using the best solution found in the problem. It should be noted that BOA is initialized with parameters including $c = 0.01$, $a = [0.1, 0.3]$ and $p = 0.5$.

IV. Results and Discussions

The investigated MPC-based MCA in Section II is developed in MATLAB software. The proposed algorithm is shown in Fig. 1. The MPC-based MCA unit, which is investigated in Section II regenerates the translational and rotational motion signal for SBMP. The computational load of the MPC model is a main concern in real-time applications. The model can be run in a real-time mode with 0.01 (s) as a time step and reach reasonable results using the KWIK [37] algorithm to find the optimal input motion signal. The control horizon and prediction horizon of the MPC are selected via the optimisation algorithms including GA and BOA to improve the motion fidelity for the driver of the SBMP. The weights of the MPC-based MCA are tuned by trial-and-error to reduce the translational and rotational motion sensation error between the SBMP user and the real vehicle.

Fig. 2.a-b illustrates the convergence of both GA and BOA algorithms in tuning the MPC horizons. The results of GA and BOA algorithm prove that the adjustment parameters of both algorithms are appropriately selected. The SBMP should respect the workspace limitation, and the rotational motion sensation error should be under the human angular threshold [15]. These are the essential requirements of the proposed method, and if there is a violation from them, the solution should be rejected and substituted by another solution. The optimised control horizon and prediction horizon of the MPC-based MCA found via GA are 450 and 9, respectively. Fig. 2.b shows the convergence of the BOA algorithm. The extracted optimised MPC horizons using BOA are 570- and 18-time steps, respectively. The convergence of the BOA algorithm is faster and better than the GA Algorithm. According to Fig. 2.a-b, BOA algorithm converges quickly and reaches a better result compared to the GA algorithm.

Fig. 3 and Table III show the results of the GA and BOA with ten times of test repetition. Fig. 3 shows the boxplot of GA and BOA with ten tests for each of them. It reveals the accuracy and repeatability of every algorithm. It proves that the BOA reaches reliable results to find the optimal MPC horizons because of their higher accuracy and repeatability.

The real motion scenario recorded via the Rigs of Rods (version 0.39.5) for the longitudinal channel, as shown in Fig. 4.a-b. Fig. 4.a shows the translational acceleration signal along *x*-axis, and Fig. 4.b shows the rotational velocity signal along *pitch*-angle for 35 seconds of the motion scenario. 0.2 translational and 0.5 rotational pre-scale factor is employed to keep the SBMP inside the workspace area [38, 39]. The three sets of control horizon and prediction horizon of the MPC-based MCA are employed to compare the results via BOA, GA and trial-and-error methods. The third way of the tuning is trial-and-error by the knowledge of the system. The control and prediction horizons of the proposed MPC-based MCA are chosen 3 and 300 time-step. The results of using three methods, including trial-and-error, GA and BOA, are shown in Table IV. The root means square error (RMSE) of the translational and rotational motion sensation error between the real vehicle and the SBMP's driver are shown in the second column of Table IV. The error of sensed angular velocity between the real and

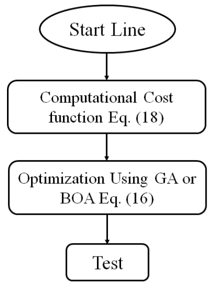

Fig. 1. The schematic structure of the proposed algorithm.

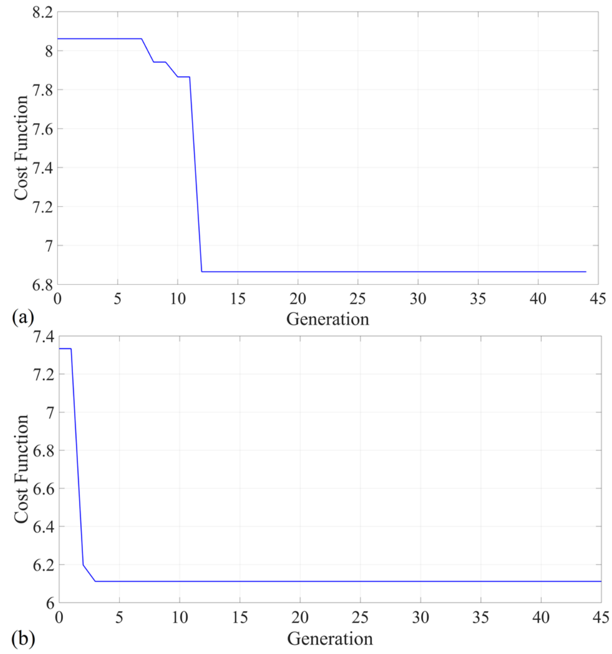

Fig. 2. The convergence of the optimization algorithms: (a) GA; (b) BOA.

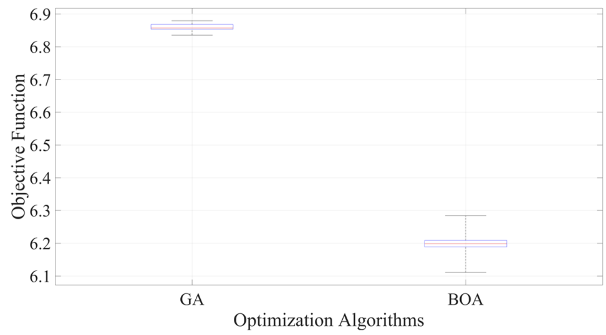

Fig. 3. The boxplots for the GA and BOA optimization algorithms with ten-time trial.

TABLE III THE COMPARISON OF OPTIMIZATION RESULTS OBTAINED FROM MATLAB MODEL OF MPC-BASED MCA

It	GA	BFO
1	6.8649	6.1106
2	6.8534	6.1947
3	6.8542	6.2837
4	6.8568	6.2003
5	6.8355	6.1885
6	6.8793	6.1106
7	6.8682	6.1955
8	6.8574	6.2034
9	6.8786	6.2837
10	6.8478	6.2082

TABLE IV THE RESULT OF THE MPC-BASED MCA USING TRIAL-AND-ERROR, GA AND BOA.

Index	RMSE			CC		
	TAE	GA	BOA	TAE	GA	BOA
SCC ω_Θ	0.0193	0.0096	0.0075	0.8140	0.9545	0.9720
OTO a_x	0.4278	0.3443	0.3348	0.1860	0.4443	0.5090

OTO: Otolith Organ; SCC: Semicircular System; TAE: Trial-and-Error; GA: Genetic Algorithm; BOA: Butterfly Optimization Algorithm; RMSE: Root Mean Square Error; CC: Correlation Coefficient;

SBMP drivers along with *pitch*-angle decrease 21.88 and 61.14 per cent using proposed MPC-based MCA via BOA tuning method compared with GA and trial-and-error, respectively. Also, the error of the sensed specific force reduces 2.76 and 21.73 per cent using the BOA method compared with GA and trial-and-error methods, respectively. Asadi et al. [40, 41] introduced a correlation coefficient (CC) for validating the motion fidelity of the regenerated motion cues. The shape similarity factor represents the CC factor in the second column of Table IV.

Fig. 5.a-b represents the actual driver and the SBMP user sensed angular velocity and sensed specific force along the *pitch*-angle and *x*-axis using MPC-based MCAs using BOA, GA and trial-and-error methods to determine control horizon and prediction horizon of MPC-based MCA. The MPC-based MCA tuning via BOA algorithm follows the reference signal with higher CC compared with tuned MPC-based MCA using GA and trial-and-error methods. The rotational and translational motion sensation using BOA to find the optimised control horizon and prediction horizon are precise with less inaccurate motion cues compared with the GA and trial-and-error methods. It makes the more convincing driving feeling by better usage of the MPC algorithm.

Fig. 6.a-b represents the error of angular and translational motion sensation between the real driver and SBMP user along the *x*-axis and *pitch*-angle using MPC-based MCAs via BOA, GA and trial-and-error methods to determine control horizon and prediction horizon of MPC-based MCA. The rotational motion sensation using the BOA to tune control horizon, and prediction horizon is precise because of following the signal accurately with higher shape similarity, as shown in Fig. 6.a. The translational motion feeling by the BOA tuned MPC-based MCA presents the better follow of the translational motion

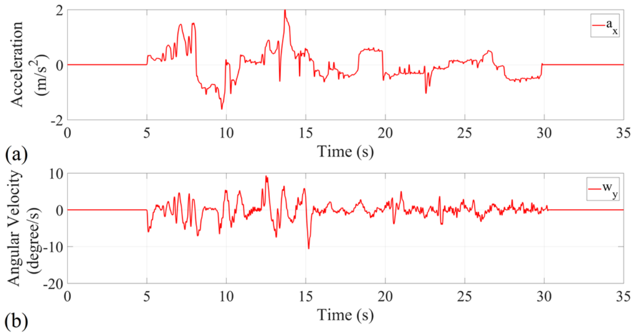

Fig. 4 The reference signal of the real driver (a): translational signal; (b): rotational signal.

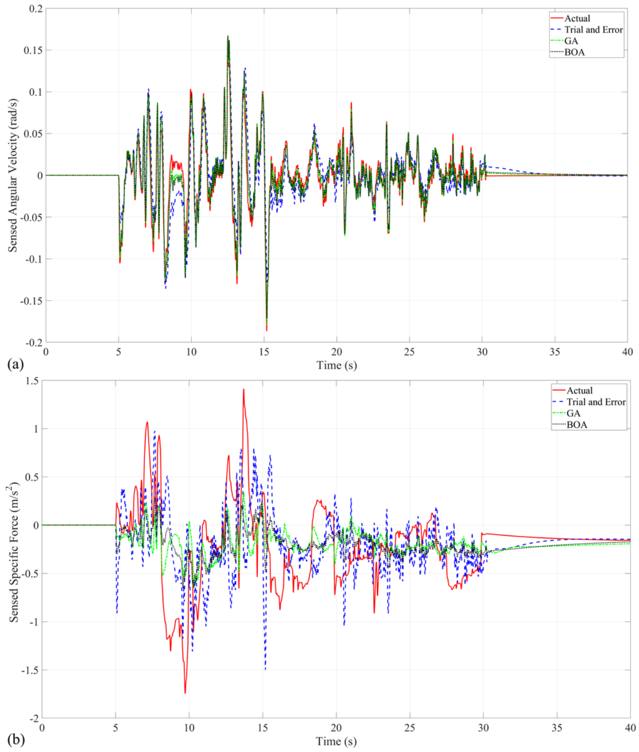

Fig. 5 The angular and translational motion sensation using the MPC-based MCA with meta-heuristic algorithm and trial-and-error tuning methods along: (a) pitch-angle; (b) x-axis.

sensation because of the capability of the best usage of the workspace boundaries, as shown in Fig. 6.b. The BOA algorithm finds the optimal control horizon and prediction horizon of the MPC-based MCA for the optimal usage of the SBMP's limitations.

Fig. 7.a-b represents the angular and linear motion of the SBMP using BOA, GA and trial-and-error tuning techniques. The proposed BOA method uses the workspace area wisely compared with GA and trial-and-error methods. Fig. 7.a-b shows that all methods respect the workspace limitations while the MPC-based MCA using BOA can reduce the motion sensation error more compared with the GA and trial-and-error tuning methods.

In this paper, BOA and GA algorithms are used to tune the

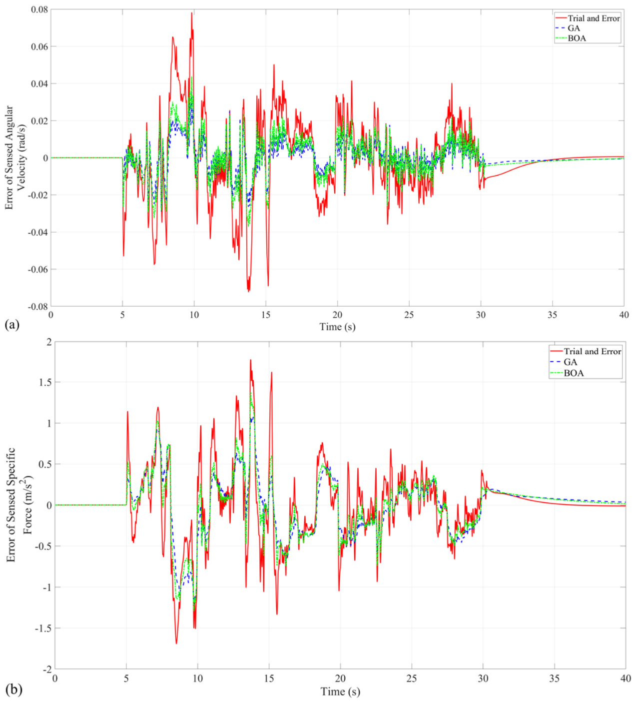

Fig. 6 The error of sensed angular velocity and specific force for the MPC-based MCAs using meta-heuristic algorithm and trial-and-error tuning methods along: (a) pitch-angle; (b) x-axis.

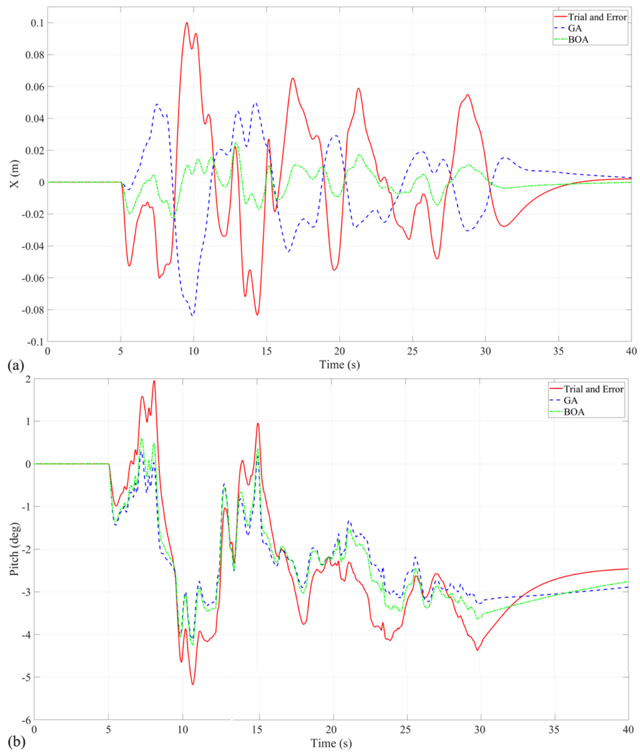

Fig. 7 The translational and rotational displacement of SBMP between the MPC-based MCA using meta-heuristic algorithm and trial-and-error tuning methods along: (a) x-axis; (b) Pitch-angle.

control horizon and prediction horizon of the MPC-based MCA. According to the outcomes, BOA finds the optimal horizons of the MPC-based MCA very quickly. Besides, the results prove the suitability of the BOA for tuning the control horizon and prediction horizon of MPC-based MCA to reproduce the high-fidelity motion cues for the SBMP user. It can regenerate the best results via MPC-based MCA.

V. Conclusion

MPC-based MCA has introduced to consider the limitations of SBMP in the regeneration of the optimal motion signal to make the similar motion sensation for the SBMP user. The MPC horizons are regularly tuned using trial-and-errors, but it cannot be the best MPC horizons to reduce the computational load of the hardware as well as the motion sensation error for the SBMP driver. The optimal and fast choice of the MPC's control and prediction horizons to find the best MCA to enhance the motion fidelity and decrease the computational load is very challenging. In this paper, GA and BOA algorithms are employed to find the optimal control horizon and prediction horizon of MPC-based MCA. The objective of this paper is to extract the optimal MPC horizons including prediction horizon and control horizon to reduce the computational load of the system, and better usage of the workspace area. The validation of the designed algorithm is accomplished using MATLAB software to show the effectiveness of the BOA tuning method compared with GA and trial-and-error. The outcomes of the work proved the efficacy of the proposed model in terms of improving the motion fidelity for the SBMP user. In the developed model, the SBMP has more capability while it respects the limitation equally. In the future study, the idea of neural network and deep learning [42-44] can be employed to introduce the real-time tuning method.

References

[1] H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 47, no. 2, pp. 238-254, 2016.

[2] A. Khatami, A. Khosravi, T. Nguyen, C. P. Lim, and S. Nahavandi, "Medical image analysis using wavelet transform and deep belief networks," *Expert Systems with Applications,* vol. 86, pp. 190-198, 2017.

[3] H. Haggag, M. Hossny, D. Filippidis, D. Creighton, S. Nahavandi, and V. Puri, "Measuring depth accuracy in RGBD cameras," in *2013, 7th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2013, pp. 1-7: IEEE.

[4] J. Iskander *et al.*, "From car sickness to autonomous car sickness: A review," *Transportation research part F: traffic psychology and behaviour,* vol. 62, pp. 716-726, 2019.

[5] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "High-Fidelity Hexarot Simulation-Based Motion Platform Using Fuzzy Incremental Controller and Model Predictive Control-Based Motion Cueing Algorithm," *IEEE Systems Journal,* 2019.

[6] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A New Gantry-Tau-Based Mechanism Using Spherical Wrist and Model Predictive Control-Based Motion Cueing Algorithm," *Robotica,* pp. 1-22, 2019.

[7] M. R. C. Qazani, S. Pedrammehr, H. Abdi, and S. Nahavandi, "Performance Evaluation and Calibration of Gantry-Tau Parallel

Mechanism," *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering,* pp. 1-15, 2019.

[8] S. Pedrammehr, M. R. C. Qazani, H. Asadi, and S. Nahavandi, "Kinematic manipulability analysis of hexarot simulators," in *The 20th IEEE International Conference on Industrial Technology, IEEE-ICIT 2019*, 2019, pp. 13-15.

[9] J.-P. Merlet, *Parallel robots*. Springer Science & Business Media, 2006.

[10] S. Pedrammehr, M. R. C. Qazani, H. Asadi, and S. Nahavandi, "Control system development of a hexarotbased high-G centrifugal simulator," in *The 20th IEEE International Conference on Industrial Technology IEEE-ICIT 2019*, 2019.

[11] M. R. C. Qazani *et al.*, "Kinematic analysis and workspace determination of hexarot-a novel 6-DOF parallel manipulator with a rotation-symmetric arm system," *Robotica,* vol. 33, no. 8, pp. 1686-1703, 2015.

[12] M. R. C. Qazani, H. Asadi, S. Pedrammehr, and S. Nahavandi, "Performance analysis and dexterity monitoring of hexapod-based simulator," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018, pp. 226-231: IEEE.

[13] M. R. C. Qazani, S. Pedrammehr, A. Rahmani, M. Shahryari, A. K. S. Rajab, and M. M. Ettefagh, "An experimental study on motion error of hexarot parallel manipulator," *The International Journal of Advanced Manufacturing Technology,* vol. 72, no. 9-12, pp. 1361-1376, 2014.

[14] H. Asadi, S. Mohamed, and S. Nahavandi, "Incorporating human perception with the motion washout filter using fuzzy logic control," *IEEE/ASME Transactions on Mechatronics,* vol. 20, no. 6, pp. 3276-3284, 2015.

[15] H. Asadi, S. Mohamed, C. P. Lim, S. Nahavandi, and E. Nalivaiko, "Semicircular canal modeling in human perception," *Reviews in the Neurosciences,* vol. 28, no. 5, pp. 537-549, 2017.

[16] H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "A review on otolith models in human perception," *Behavioural brain research,* vol. 309, pp. 67-76, 2016.

[17] P. R. MacNeilage, M. S. Banks, D. R. Berger, and H. H. Bülthoff, "A Bayesian model of the disambiguation of gravitoinertial force by visual cues," *Experimental Brain Research,* vol. 179, no. 2, pp. 263-290, 2007.

[18] B. Conrad and S. F. Schmidt, "A study of techniques for calculating motion drive signals for flight simulators," 1971.

[19] S. F. Schmidt, *Motion drive signals for piloted flight simulators*. National Aeronautics and Space Administration, 1970.

[20] H. Asadi, C. P. Lim, S. Mohamed, D. Nahavandi, and S. Nahavandi, "Increasing Motion Fidelity in Driving Simulators Using a Fuzzy-Based Washout Filter," *IEEE Transactions on Intelligent Vehicles,* vol. 4, no. 2, pp. 298-308, 2019.

[21] H. Asadi, S. Mohamed, D. Rahim Zadeh, and S. Nahavandi, "Optimisation of nonlinear motion cueing algorithm based on genetic algorithm," *Vehicle System Dynamics,* vol. 53, no. 4, pp. 526-545, 2015.

[22] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, "Model-based predictive motion cueing strategy for vehicle driving simulators," *Control Engineering Practice,* vol. 17, no. 9, pp. 995-1003, 2009.

[23] Z. Fang and A. Kemeny, "Explicit MPC motion cueing algorithm for real-time driving simulator," in *Proceedings of The 7th International Power Electronics and Motion Control Conference*, 2012, vol. 2, pp. 874-878: IEEE.

[24] M. Bruschetta, F. Maran, and A. Beghi, "A nonlinear, mpc-based motion cueing algorithm for a high-performance, nine-dof dynamic simulator platform," *IEEE Transactions on Control Systems Technology,* vol. 25, no. 2, pp. 686-694, 2016.

[25] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit—An open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods,* vol. 32, no. 3, pp. 298-312, 2011.

[26] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A decoupled linear model predictive control-based motion cueing algorithm for simulation-based motion platform with limitted workspace," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC), Bari, Italy*, 2019, pp. 1-6.

[27] M. R. C. Qazani, H. Asadi, S. Khoo, and S. Nahavandi, "A Linear Time-Varying Model Predictive Control-Based Motion Cueing Algorithm for Hexapod Simulation-Based Motion Platform," *IEEE Transactions on Systems, Man, and Cybernetic: Systems,* 2019.

[28] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A Model Predictive Control-Based Motion Cueing Algorithm with Consideration of Joints' limitations for Hexapod Motion Platform," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 708-713: IEEE.

[29] H. Asadi, S. Mohamed, K. Nelson, S. Nahavandi, and D. R. Zadeh, "Human perception-based washout filtering using genetic algorithm," in *International Conference on Neural Information Processing*, 2015, pp. 401-411: Springer.

[30] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Multiobjective and interactive genetic algorithms for weight tuning of a model predictive control-based motion cueing algorithm," *IEEE transactions on cybernetics,* vol. 49, no. 9, pp. 3471-3481, 2018.

[31] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Optimizing model predictive control horizons using genetic algorithm for motion cueing algorithm," *Expert Systems with Applications,* vol. 92, pp. 73-81, 2018.

[32] K. I. Kouramas, N. P. Faísca, C. Panos, and E. N. Pistikopoulos, "Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming," *Automatica,* vol. 47, no. 8, pp. 1638-1645, 2011.

[33] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[34] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "OpenGA, a C++ genetic algorithm library," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2051-2056: IEEE.

[35] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing,* vol. 23, no. 3, pp. 715-734, 2019.

[36] W.-Y. Lin, W.-Y. Lee, and T.-P. Hong, "Adapting crossover and mutation rates in genetic algorithms," *J. Inf. Sci. Eng.,* vol. 19, no. 5, pp. 889-903, 2003.

[37] C. Schmid and L. T. Biegler, "Quadratic programming methods for reduced hessian SQP," *Computers & chemical engineering,* vol. 18, no. 9, pp. 817-832, 1994.

[38] H. Asadi, C. P. Lim, A. Mohammadi, S. Mohamed, S. Nahavandi, and L. Shanmugam, "A genetic algorithm–based nonlinear scaling method for optimal motion cueing algorithm in driving simulator," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 232, no. 8, pp. 1025-1038, 2018.

[39] H. Asadi *et al.*, "A Model Predictive Control-based Motion Cueing Algorithm using an optimized Nonlinear Scaling for Driving Simulators," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 1245-1250: IEEE.

[40] H. Asadi, "Human perception-based washout filtering," Deakin Univeristy2015.

[41] H. Asadi, S. Mohamed, K. Nelson, S. Nahavandi, and M. Oladazimi, "An optimal washout filter based on genetic algorithm compensators for improving simulator driver perception," in *DSC 2015: Proceedings of the Driving Simulation Conference & Exhibition*, 2015, pp. 1-10: Max Planck Institute for the Advancement of Science.

[42] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE access,* vol. 6, pp. 36218-36234, 2018.

[43] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Partial adversarial training for prediction interval," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1-6: IEEE.

[44] H. M. D. Kabir and S. M. Salahuddin, "A frequency multiplier using three ambipolar graphene transistors," *Microelectronics journal,* vol. 70, pp. 12-15, 2017.