Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 415 (2023) 116261
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Machine-learning a perfect bending soccer goal shot

T.I. Zohdi

Department of Mechanical Engineering, 6117 Etcheverry Hall, University of California, Berkeley, CA, 94720-1740, USA

Received 22 June 2023; received in revised form 5 July 2023; accepted 5 July 2023
Available online 3 August 2023

Abstract

The objective of this work is to ascertain the optimal bending kick velocity and spin that a player should impart to exactly hit a target within a soccer goal, using machine-learning optimization. Specifically, the work develops a model of a kicking player producing a high-velocity spinning soccer ball which interacts with the surrounding air that induces drag forces and the Magnus effect, both of which are functions of the Reynolds number. This yields a set of highly nonlinear, coupled, differential equations. The framework is designed to enable digital-twin type technologies, i.e. digital replicas that run in real time with the physical system on laptops or other mobile systems for rapid feedback. The overall guiding motivation is to provide a useful tool to assist coaches and to rapidly train players. Numerical examples are provided to illustrate the process.
© 2023 Elsevier B.V. All rights reserved.

Keywords: Soccer; Kicks; Goal optimization; Machine-learning

1. Introduction

Soccer is the most popular sport on Earth. One of the most fascinating shots in the sport is a high-velocity bending shot on goal, which is induced by the spin a player imparts on the ball. Probably the most famous goal harnessing this effect was the 1997 free kick of R. Carlos (https://www.youtube.com/watch?v=crKwlbwvr88, [1], which is arguably the greatest kick in the history of the game. The effect that bends the shot is referred to as the Magnus effect, and is due to a pressure differential between the points on the ball whose relative spinning action moves in the same direction as the velocity of the surrounding fluid and the relative spinning action of points that move in the opposite direction of the velocity of the surrounding fluid (Fig. 1). The detailed physics behind the Magnus effect are well-explored (see [1],[2], [3], [4], [5], [6], [7], [8], [9], [10], [11]). The objective of this work is to ascertain the optimal kick velocity, direction and spin that a player should impart to exactly hit within a target soccer goal with a bending kick, using machine-learning optimization. Specifically, the work develops a model of a kicking player and the subsequent high-velocity spinning soccer ball, interacting with the surrounding air which induces drag forces and the Magnus effect, which are both functions of the Reynolds number. This yields a set of highly nonlinear, coupled differential equations. The framework is designed to enable digital-twin type technologies, i.e. digital replicas that run in real time with the physical system on laptops or other mobile systems for rapid feedback. The overall guiding motivation is to provide a useful tool to rapidly train players. Numerical examples are provided to illustrate the process.

E-mail address: zohdi@berkeley.edu.

https://doi.org/10.1016/j.cma.2023.116261
0045-7825/© 2023 Elsevier B.V. All rights reserved.

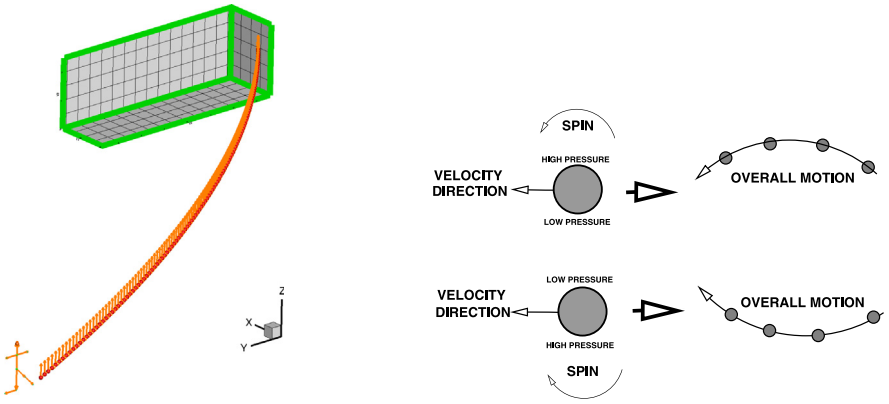

Fig. 1. The basic action of the Magnus effect. The effect that bends the shot is due to a pressure differential between the points whose relative spinning action move in the same direction as the velocity of the surrounding fluid and the relative spinning action that moves points move in the opposite direction to the velocity of the surrounding fluid. On the left, frames of a kick simulation yielding a precise target goal are shown. The optimization of was a byproduct of machine-learning algorithm developed later in this work. The arrows indicate the direction of the axis of spin.

2. Model

We assume that the soccer ball is spherical, with radius R and mass m. From Newton's laws, a force balance yields the following differential equation

$$m\dot{\boldsymbol{v}} = \boldsymbol{\Psi}^{grav} + \boldsymbol{\Psi}^{drag} + \boldsymbol{\Psi}^{Magnus}, \tag{2.1}$$

where three main forces are (a) gravity, (b) drag and (c) the Magnus force from spinning. We make the following assumptions:

- For the drag, we will employ a general phenomenological model

$$\boldsymbol{\Psi}^{drag} = \frac{1}{2}\rho_a C_D \|\boldsymbol{v}^f - \boldsymbol{v}\|(\boldsymbol{v}^f - \boldsymbol{v})A, \tag{2.2}$$

  where C_D is the drag coefficient, A is the reference area, which for a sphere is $A = \pi R^2$, R being the radius, ρ_a is the density of the ambient fluid environment and $\boldsymbol{v}^f$ is the velocity of the surrounding medium which, in the case of interest, is air. We will assume that the velocity of the surrounding fluid medium ($\boldsymbol{v}^f$) is given, implicitly assuming that the dynamics of the surrounding medium is unaffected by the ball.[1]
- For the Magnus effect

$$\boldsymbol{\Psi}^{Magnus} = \rho_a C_L A R \boldsymbol{\omega} \times (\boldsymbol{v} - \boldsymbol{v}^f), \tag{2.3}$$

  where $\boldsymbol{\omega}$ is the angular velocity of the ball, the lift coefficient correlates in the following manner for moderate Reynolds numbers (see Bush [2])

$$C_L = \frac{\|\boldsymbol{\omega}\| R}{\|\boldsymbol{v} - \boldsymbol{v}^f\|}. \tag{2.4}$$

From Newton's laws, this yields the following differential equation

$$\begin{aligned} m\dot{\boldsymbol{v}} &= \boldsymbol{\Psi}^{grav} + \boldsymbol{\Psi}^{drag} + \boldsymbol{\Psi}^{Magnus} \\ &= m\boldsymbol{g} + \frac{1}{2}\rho_a C_D \|\boldsymbol{v}^f - \boldsymbol{v}\|(\boldsymbol{v}^f - \boldsymbol{v})A + \rho_a \frac{\|\boldsymbol{\omega}\| R}{\|\boldsymbol{v} - \boldsymbol{v}^f\|} A R \boldsymbol{\omega} \times (\boldsymbol{v} - \boldsymbol{v}^f). \end{aligned} \tag{2.5}$$

In order to accurately model the effects of drag, one can take into account that the empirical drag coefficient varies with Reynolds number. For example, consider the following piecewise relation (Chow [12]):

[1] We will discuss these assumptions further, later in the work.

- For $0 < Re \leq 1$, $C_D = \frac{24}{Re}$,
- For $1 < Re \leq 400$, $C_D = \frac{24}{Re^{0.646}}$,
- For $400 < Re \leq 3 \times 10^5$, $C_D = 0.5$,
- For $3 \times 10^5 < Re \leq 2 \times 10^6$, $C_D = 0.000366 Re^{0.4275}$ and
- For $2 \times 10^6 < Re < \infty$, $C_D = 0.18$,

where the local Reynolds number for a ball is $Re \stackrel{\text{def}}{=} \frac{2R\rho_a \|\boldsymbol{v}^f - \boldsymbol{v}\|}{\mu_f}$ and μ_f is the fluid viscosity. *We note that in the zero Reynolds number limit, the drag is Stokesian.*

3. Numerical solution

The resulting set of nonlinear, highly coupled differential equations in Eq. (2.5) requires temporal discretization, where we integrate the velocity numerically

$$\begin{aligned} \boldsymbol{v}(t+\Delta t) &= \boldsymbol{v}(t) + \frac{1}{m}\int_t^{t+\Delta t} (\boldsymbol{\Psi}^{grav} + \boldsymbol{\Psi}^{drag} + \boldsymbol{\Psi}^{Magnus})\, dt \\ &\approx \boldsymbol{v}(t) + \frac{\Delta t}{m}\left(\boldsymbol{\Psi}^{grav}(t) + \boldsymbol{\Psi}^{drag}(t) + \boldsymbol{\Psi}^{Magnus}(t)\right). \end{aligned} \tag{3.1}$$

The position is the obtained by integrating again:

$$\boldsymbol{r}(t+\Delta t) = \boldsymbol{r}(t) + \int_t^{t+\Delta t} \boldsymbol{v}(t)\, dt \approx \boldsymbol{r}(t) + \Delta t \boldsymbol{v}(t). \tag{3.2}$$

Remark 1. There are a large number of physically similar phenomena relating to solid–fragment–fluid dynamics associated with blasts, explosions and fire embers. We refer the interested reader to the wide array of literature on this topic; see Plimpton [13], Brock [14], Russell [15], Shimanzu [16], Werrett [17], Kazuma [18,19], Wingerden et al. [20] and Fernandez-Pello [21], Pleasance and Hart [22], Stokes [23] and Rowntree and Stokes [24], Hadden et al. [25] and Urban et al. [26]. This numerical approach has been used repeatedly for a variety of physically similar problems in Zohdi [27–30].

Remark 2. The piecewise drag law of Chow [12] is a mathematical description for the Reynolds number over a wide range and is a curve-fit of extensive data from Schlichting [31].

Remark 3. More detailed analysis of the solid–fluid interaction between the ball and surrounding fluid would need to employ high-fidelity models requiring discretization of the Navier–Stokes equations:

$$\begin{aligned} &Balance\ of\ mass: \frac{\partial \rho}{\partial t} = -\nabla_x \rho \cdot \boldsymbol{v} - \rho \nabla_x \cdot \boldsymbol{v}, \\ &Balance\ of\ momentum: \rho\left(\frac{\partial \boldsymbol{v}}{\partial t} + (\nabla_x \boldsymbol{v}) \cdot \boldsymbol{v}\right) = \nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f}, \end{aligned} \tag{3.3}$$

where $\rho(\boldsymbol{x})$ is the density field of the fluid, $\boldsymbol{v}(\boldsymbol{x})$ is the fluid velocity field, $\boldsymbol{\sigma}(\boldsymbol{x})$ is the fluid stress field and $\boldsymbol{f}(\boldsymbol{x})$ is the body force field. Such models are significantly more complex than the models used in the current work, and would require spatio-temporal discretization solid–fluid interaction schemes, treating the ball as a part of the fluid continuum (as another fluid or solid phase), and thus meshing them in a detailed manner. However, while such models may be necessary in many industrial applications where high precision is required, for the objectives of this work, namely the overall motion of the ball, it is unwarranted.

4. Machine-learning for system optimization

The rapid rate at which simulations of the previous equations can be completed, enables the ability to explore inverse problems seeking to determine what parameter combinations can deliver a desired result (Figs. 2). In order to cast the objective mathematically, we set the problem up as a Machine Learning Algorithm (MLA); specifically a Genetic Algorithm (GA) variant, which is well-suited for nonconvex optimization. Following Zohdi [32], [33], [34],

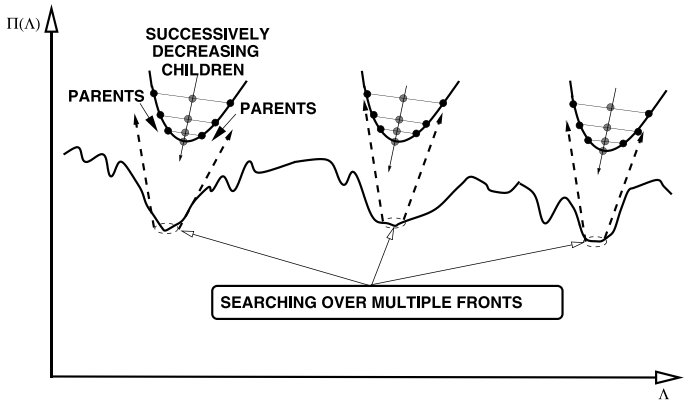

Fig. 2. The basic multi-frontal search induced by a MLA/GA-Machine Learning Algorithm/Genetic Algorithm (Zohdi [32], [33], [34], [35], [36], [37]).

[35], [36], [37], we formulate the objective as a cost function minimization problem that seeks system parameters that match a desired response

$$\Pi(\lambda_1, \ldots \lambda_8) = w_1 \frac{\|\boldsymbol{r} - \boldsymbol{T}\|}{R} + w_2 \gamma, \tag{4.1}$$

where w_1 and w_2 are weights, $\boldsymbol{r}$ is the position of the ball, $\boldsymbol{T}$ is the position of the target within the goal and γ is an indicator function where (1) if the ball is in the goal, $\gamma = 0$ and where (2) if the ball is outside of the goal, $\gamma = 1$. We systematically minimize Remark 6, $min_{\Lambda} \Pi$, by varying the design parameters: $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \ldots, \Lambda_N^i\} \stackrel{\text{def}}{=} \{velocity, directions, spin, \ldots\}$. The system parameter search is conducted within the constrained ranges of $\Lambda_1^{(-)} \leq \Lambda_1 \leq \Lambda_1^{(+)}$, $\Lambda_2^{(-)} \leq \Lambda_2 \leq \Lambda_2^{(+)}$ and $\Lambda_3^{(-)} \leq \Lambda_3 \leq \Lambda_3^{(+)}$, etc. These upper and lower limits would, in general, be dictated by what is physically feasible.

4.1. Machine-learning algorithm (MLA)

Cost functions such as Π are nonconvex in design parameter space and often nonsmooth. Their minimization is usually difficult with direct application of gradient-based methods. This motivates nonderivative search methods, for example those found in machine-learning algorithms (MLAs). One of the most basic subsets of MLAs are so-called Genetic Algorithms (GAs). For a review of GAs, see the pioneering work of John Holland ([38,39]), as well as Goldberg [40], Davis [41], Onwubiko [42] and Goldberg and Deb [43]. A description of the algorithm will be described next, following Zohdi [32], [33], [34], [35], [36], [37].

4.2. Algorithmic structure

The MLA/GA approach is extremely well-suited for nonconvex, nonsmooth, multicomponent, multistage systems and, broadly speaking, involves the following essential concepts (Fig. 2):

1. **POPULATION GENERATION:** Generate a parameter population of genetic strings: $\boldsymbol{\Lambda}^i$
2. **PERFORMANCE EVALUATION:** Compute performance of each genetic string: $\Pi(\boldsymbol{\Lambda}^i)$
3. **RANK STRINGS:** Rank them $\boldsymbol{\Lambda}^i$, $i = 1, \ldots, S$ from best to worst
4. **MATING PROCESS:** Mate pairs/produce offspring
5. **GENE ELIMINATION:** Eliminate poorly performing genetic strings
6. **POPULATION REGENERATION:** Repeat process with updated gene pool and new *random* genetic strings
7. **SOLUTION POST-PROCESSING:** Employ gradient-based methods afterwards in local "valleys"-*if smooth enough*

4.3. Specifics

Following Zohdi [32], [33], [34], [35], [36], [37], the algorithm is as follows:

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^i, (i = 1, 2, 3, \ldots, S)$:

$$\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \Lambda_1^i \\ \Lambda_2^i \\ \Lambda_3^i \\ \cdots \\ \Lambda_N^i \end{array} \right\} \tag{4.2}$$

- **STEP 2:** Compute fitness of each string $\Pi(\boldsymbol{\Lambda}^i)$, (i=1, ..., S)
- **STEP 3:** Rank genetic strings: $\boldsymbol{\Lambda}^i$, (i=1, ..., S) from best to worst
- **STEP 4:** Mate nearest pairs and produce two offspring, (i=1, ..., S):

$$\boldsymbol{\lambda}^i \stackrel{\text{def}}{=} \boldsymbol{\Phi} \circ \boldsymbol{\Lambda}^i + (\mathbf{1} - \boldsymbol{\Phi}) \circ \boldsymbol{\Lambda}^{i+1} \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \phi_1 \Lambda_1^i \\ \phi_2 \Lambda_2^i \\ \phi_3 \Lambda_3^i \\ \cdots \\ \phi_N \Lambda_N^i \end{array} \right\} + \left\{ \begin{array}{c} (1-\phi_1)\Lambda_1^{i+1} \\ (1-\phi_2)\Lambda_2^{i+1} \\ (1-\phi_3)\Lambda_3^{i+1} \\ \cdots \\ (1-\phi_N)\Lambda_N^{i+1} \end{array} \right\} \tag{4.3}$$

and

$$\boldsymbol{\lambda}^{i+1} \stackrel{\text{def}}{=} \boldsymbol{\Psi} \circ \boldsymbol{\Lambda}^i + (\mathbf{1} - \boldsymbol{\Psi}) \circ \boldsymbol{\Lambda}^{i+1} \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \psi_1 \Lambda_1^i \\ \psi_2 \Lambda_2^i \\ \psi_3 \Lambda_3^i \\ \cdots \\ \psi_N \Lambda_N^i \end{array} \right\} + \left\{ \begin{array}{c} (1-\psi_1)\Lambda_1^{i+1} \\ (1-\psi_2)\Lambda_2^{i+1} \\ (1-\psi_3)\Lambda_3^{i+1} \\ \cdots \\ (1-\psi_N)\Lambda_N^{i+1} \end{array} \right\} \tag{4.4}$$

where for this operation, the ϕ_i and ψ_i are random numbers, such that $0 \leq \phi_i \leq 1$, $0 \leq \psi_i \leq 1$, which are different for each component of each genetic string

- **STEP 5:** Eliminate the bottom M strings and keep top K parents and their K offspring (K offspring+K parents+$M = S$)
- **STEP 6:** Repeat STEPS 1–5 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings
- **REFOCUS OPTION:** One can refocus search around best performing parameter set every few generations, thus concentrating the computation effort around the most promising (optimal) areas of design space. This operation is performed by re-centering the search around the best genetic string and performing the search in a new constrained local interval about that best genetic string, then proceeding to STEP 1, as before.

Remark 4. If one selects the mating parameters $\phi's$ and $\psi's$ to be greater than one and/or less than zero, one can induce "mutations", i.e. characteristics that neither parent possesses. However, this is somewhat redundant with introduction of new random members of the population in the current algorithm. If one does not retain the parents in the algorithm above, it is possible that inferior performing offspring may replace superior parents. Thus, top parents should be kept for the next generation. This guarantees a monotone reduction in the cost function, unless one implements the refocusing option mentioned previously. Retained parents do not need to be reevaluated, making the algorithm less computationally expensive, since these parameter sets do not have to be reevaluated in the next generation's calculations. In the absence of refocussing, numerous studies of the author (Zohdi [32], [33], [34], [35], [36], [37]) have shown that the advantages of parent retention outweighs inbreeding, for sufficiently large population sizes. Finally, we observe that this algorithm is easy to parallelize.

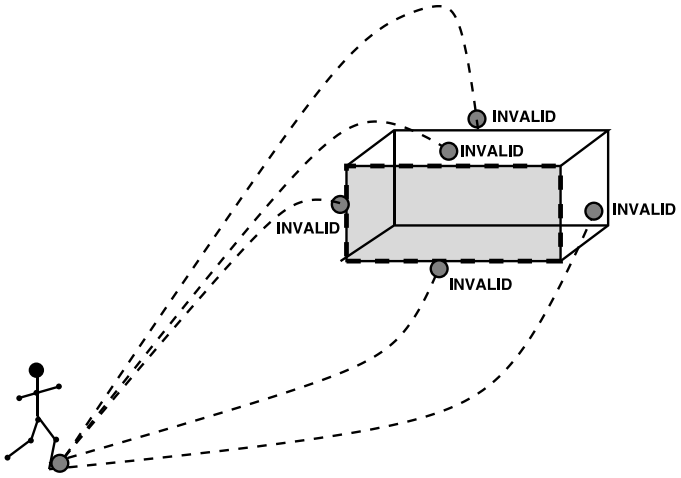

Fig. 3. The unilateral constraints to disallow improper penetration of the goal from the top, sides, bottom or back.

Remark 5. After application of such a global search algorithm, one can apply a gradient-based method around the best performing parameter set, if the objective function is sufficiently smooth in that region of the parameter space. In other words, if one has located a convex portion of the parameter space with a global genetic search, one can employ gradient-based procedures locally to minimize the objective function further, since they are generally much more efficient for convex optimization of smooth functions. An exhaustive review of these methods can be found in the texts of Luenberger [44] and Gill, Murray and Wright [45]. *However, refocussing usually makes this extra step unnecessary, since the search eventually concentrates the computational effort locally around the best parameter set beforehand.*

Remark 6. Unilateral constraints are enforced to disallow improper penetration of the goal from the top, sides, bottom or back (referred to as invalid planar boundaries, Fig. 3). If the ball hits the ground or the *invalid* planar boundaries or if the ball enters the goal in a valid manner, the simulation is stopped: $\boldsymbol{v}(t) = \boldsymbol{0}$ and the cost function in Equation is assessed.

4.4. Model problem-parameter settings

We applied the MLA algorithm, with the following parameters (eight):

- Λ_1 =the ball's initial velocity (magnitude),
- $\Lambda_2, \Lambda_3, \Lambda_4$ = the ball's velocity direction,
- Λ_5 =the ball's angular velocity (magnitude),
- $\Lambda_6, \Lambda_7, \Lambda_8$ = the ball's angular velocity direction.

Explicitly, the design vector is:

$$\Lambda = \{\|\boldsymbol{v}(0)\|, n_1, n_2, n_3, \|\boldsymbol{\omega}(0)\|, n_1, n_2, n_3\}. \tag{4.5}$$

The search ranges were

- 30 m/s $= \Lambda_1^- \leq \Lambda_1^- \leq \Lambda_1^+ = 70$ m/s,
- $(-1, -1, -1) = (\Lambda_2^-, \Lambda_3^-, \Lambda_4^-) \leq (\Lambda_2, \Lambda_3, \Lambda_4) \leq (\Lambda_2^+, \Lambda_3^+, \Lambda_4^+) = (1, 1, 1)$,
- 50 rad/s $= \Lambda_5^- \leq \Lambda_5^- \leq \Lambda_5^+ = 150$ rad/s,
- $(-1, -1, -1) = (\Lambda_6^-, \Lambda_7^-, \Lambda_8^-) \leq (\Lambda_6, \Lambda_7, \Lambda_8) \leq (\Lambda_6^+, \Lambda_7^+, \Lambda_8^+) = (1, 1, 1)$,

which of course will change if the search re-adaptation option is chosen. Also note that the direction components are normalized during the search to yield a unit normal vector $\frac{\boldsymbol{n}}{\|\boldsymbol{n}\|}$. Additionally, we used total time $T = 0.5$ seconds (see Fig. 4). .

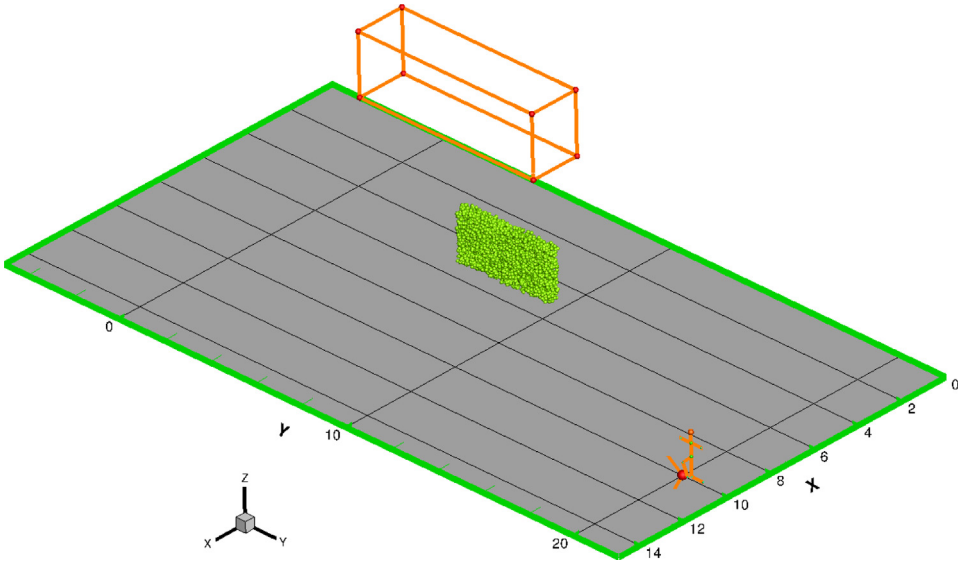

Fig. 4. Model problem for optimization of a kick avoiding the wall. The ball was initially located at $\boldsymbol{r}(0) = \{10, 20, 0\}$, and goal centerline at $\{0, 0, 0\}$, and the target being in the upper left corner of the goal $\boldsymbol{T} = \{0, -3, 2.3\}$. We also inserted a wall located at $\{5, 7.5, 0\}$ that was 0.5 m thick, 4 m wide and 2 m tall.

4.4.1. Numerical results

In order to illustrate the model, the following simulation parameters were chosen (Figs. 5–7):

- The size of the ball $R = 0.11$m with mass $m = 0.43$ kg (FIFA standard).
- The goal dimensions are 7.32 m wide and 2.44 m in height (FIFA standard).
- The ball was initially located at $\boldsymbol{r}(0) = \{10, 20, 0\}$, and goal centerline at $\{0, 0, 0\}$, and the target being in the upper left corner of the goal $\boldsymbol{T} = \{0, -3, 2.3\}$.
- The density of air, $\rho_a = 1.225$, kg/m^3.
- The viscosity for air, $\mu_f = 0.000018$ Pa-s.
- The surrounding fluid velocity was set to zero ($\boldsymbol{v}^f = \boldsymbol{0}$).
- Total simulation duration, 0.5 s with time step size, Δt= 0.001 s.
- The weights in the cost function in Remark 6 were $w_1 = w_2 = 1$.

We also inserted a wall located at $\{5, 7.5, 0\}$ that was 0.5 m thick, 4 m wide and 2 m tall. The design parameters $\boldsymbol{\Lambda} = \{\Lambda_1, \Lambda_2 ... \Lambda_N\}$ were optimized over the search intervals (8 variables): $\boldsymbol{\Lambda}_i^- \leq \boldsymbol{\Lambda} \leq \boldsymbol{\Lambda}_i^+$, $i = 1, 2, \ldots, 8$. We used the following MLA settings:

- Number of design variables: 8,
- Population size per generation: 24,
- Number of parents to keep in each generation: 6,
- Number of children created in each generation: 6,
- Number of completely new genes created in each generation: 12,
- Number of generations for re-adaptation around a new search interval: 10 and
- Number of generations: 500.

The algorithm was automatically reset every *10 generations*. The entire 500 generation simulation, with 24 genes per evaluation (12000 total designs) took approximately 0.25 s on a Macbook Pro laptop, *making it ideal as a design tool.* Fig. 8 (average population of 24 genes performance and top gene performance) and Table 1 (values of the gene components) illustrate the results. Fig. 8 shows reduction of the cost function for the 8 parameter set. Allowing the MLA/GA to readapt every 10 generations, leads to the nonmonotone reduction of the cost function. Shown are the best performing gene (design parameter set, in *red*) as a function of successive generations, as well

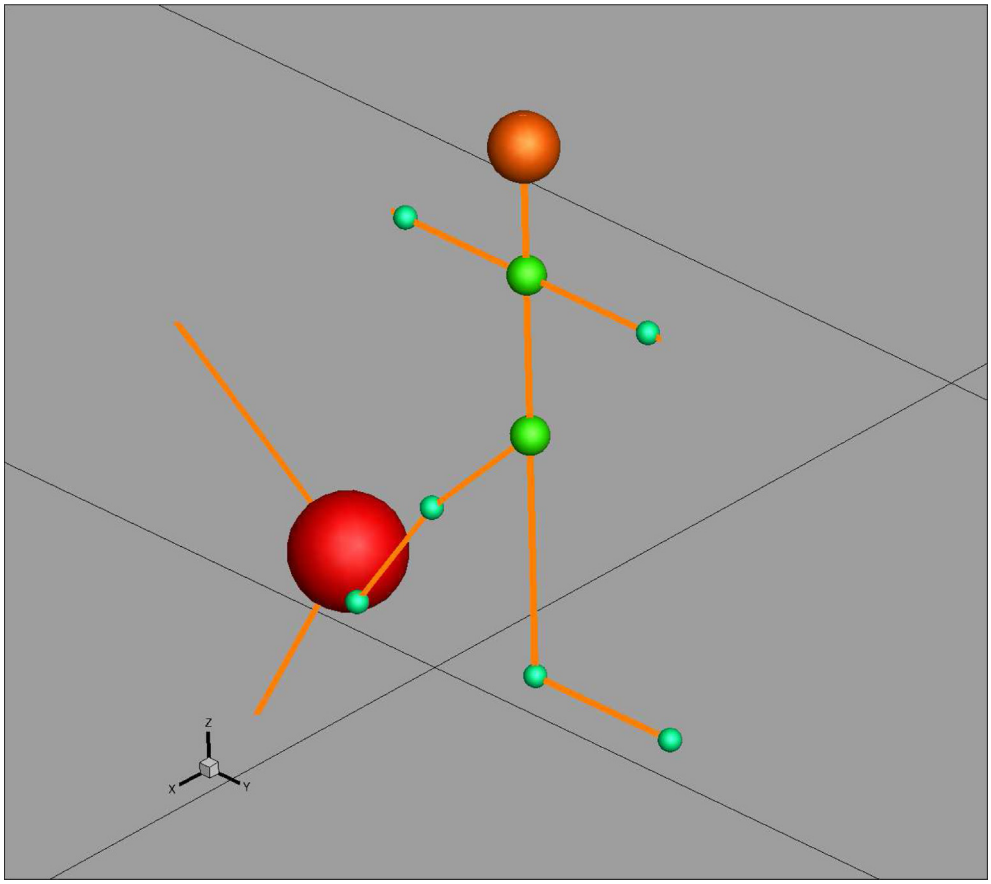

Fig. 5. Zoom on model problem for optimization of a kick avoiding the wall. The ball was initially located at $\boldsymbol{r}(0) = \{10, 20, 0\}$, and goal centerline at $\{0, 0, 0\}$, and the target being in the upper left corner of the goal $\boldsymbol{T} = \{0, -3, 2.3\}$. We also inserted a wall located at $\{5, 7.5, 0\}$ that was 0.5 m thick, 4 m wide and 2 m tall.

Table 1
The top system parameter performers ($\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_8$).

$\boldsymbol{\Lambda}_1$	$\boldsymbol{\Lambda}_2$	$\boldsymbol{\Lambda}_3$	$\boldsymbol{\Lambda}_4$	$\boldsymbol{\Lambda}_5$	$\boldsymbol{\Lambda}_6$	$\boldsymbol{\Lambda}_7$	$\boldsymbol{\Lambda}_8$	Π
56.2833	−0.3043	−0.9282	0.2287	146.8512	0.6590	−0.0631	−0.7495	0.1158

as the average performance of the entire population of genes (designs, in *green*). The optimal design is extremely tight after approximately 260 generations, and any deviation from the optimal leads to the wall stopping the ball (as exhibited by the green population average). On the right of Fig. 8, the reduction of the cost function for the 8 parameter set *without the wall present*. In that case, the optimal design is *not* extremely tight after approximately 260 generations, relative to the simulation with the wall, hence the insensitivity of the population average. The cost function (error) was massively reduced to $\Pi = 0.1158$, which implies

$$\Pi(\lambda_1, \ldots \lambda_8) = \frac{\|\boldsymbol{r} - \boldsymbol{T}\|}{R} = 0.1158 \Rightarrow \|\boldsymbol{r} - \boldsymbol{T}\| = 0.1158R, \tag{4.6}$$

or with a ball radius of $R = 0.11$ m, an error distance of $\|\boldsymbol{r} - \boldsymbol{T}\| = 0.0127$ m, i,.e. *approximately a 1.27 cm difference between the ball center and the exact target, from a shot of over 20 m away.* In Fig. 6, from left to right and top to bottom, frames of a kick simulation yielding a precise target goal, using the machine-learning algorithm. The arrows indicate the direction of the velocity and the axis of spin. In summary, Figs. 5–7 illustrate the power of the approach to ascertain extremely complex parameter sets for optimal training purposes.

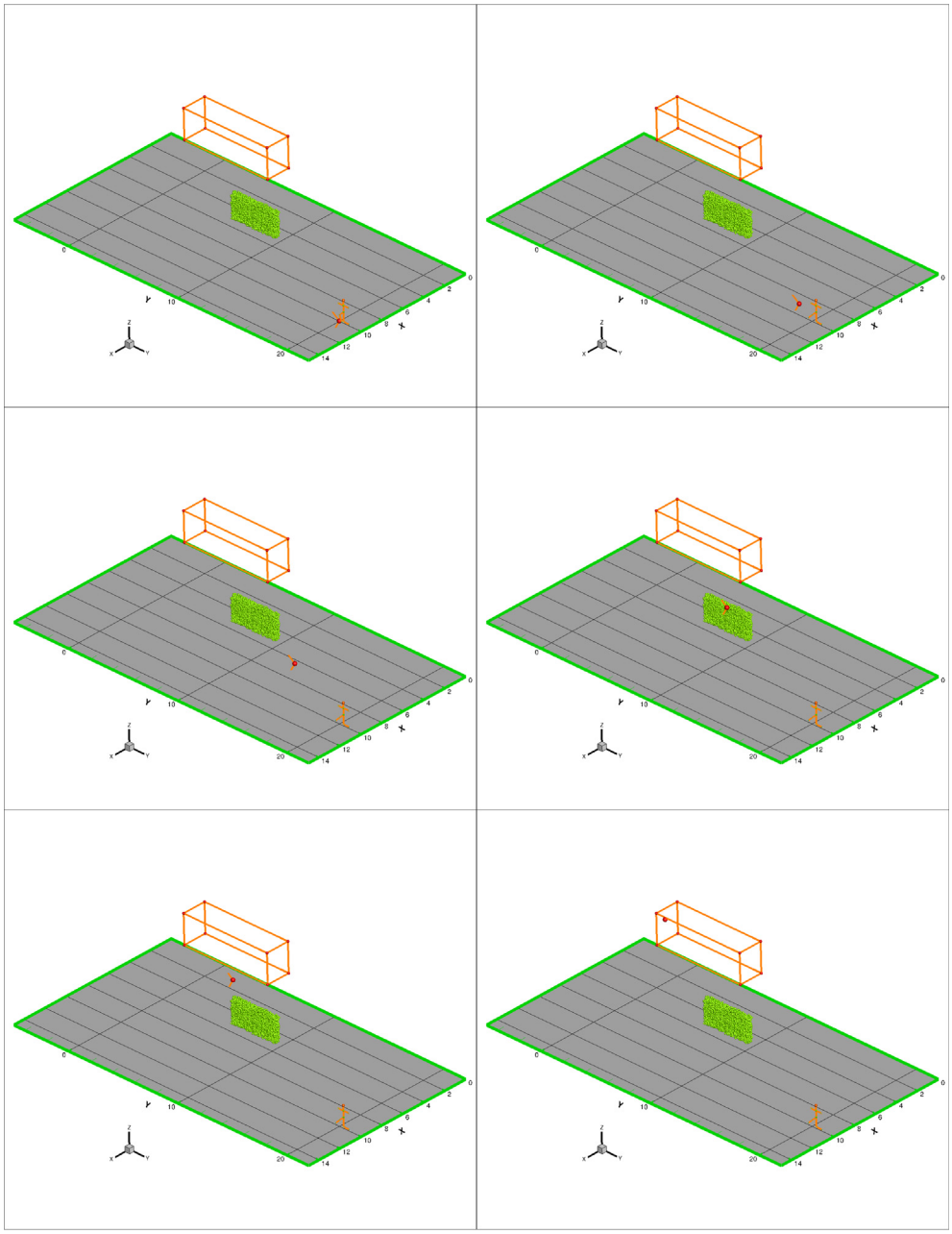

Fig. 6. From left to right and top to bottom: frames of a kick simulation yielding a precise target goal. The optimization of was a byproduct of the machine-learning algorithm developed in this work. The arrows indicate the direction of the velocity and the axis of spin.

4.5. An example of instructions to players

In order to provide actionable instructions to player (Fig. 5), one can post-process the results by utilizing basic kinematics, namely the velocity relationship between the ball center of mass and a possible point of contact ($\boldsymbol{r}_p$, Fig. 9)

$$\boldsymbol{v}_p = \boldsymbol{v}_{cm} + \boldsymbol{v}_{cm\to p} = \boldsymbol{v}_{cm} + \boldsymbol{\omega} \times \boldsymbol{r}_{cm\to p}. \tag{4.7}$$

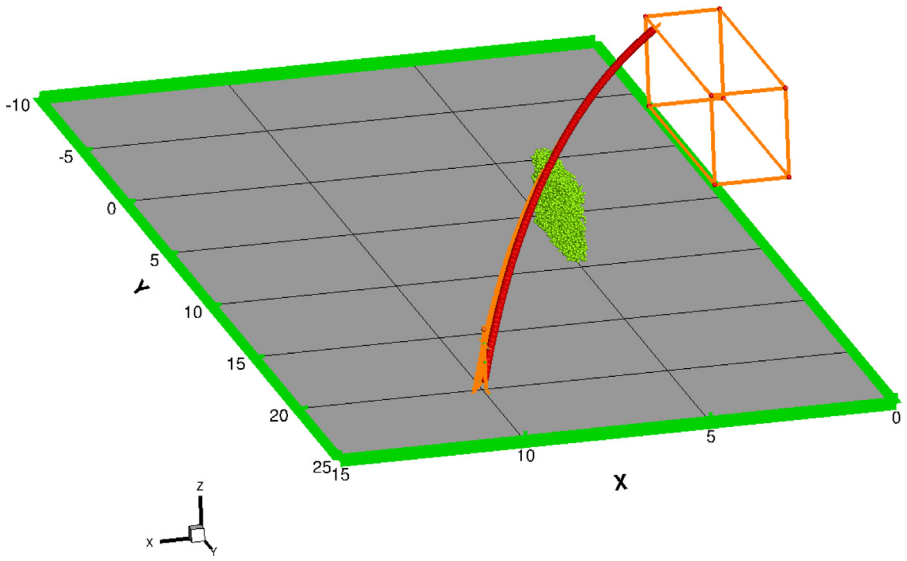

Fig. 7. Result of the model problem for optimization of a kick avoiding the wall. The ball was initially located at $\boldsymbol{r}(0) = \{10, 20, 0\}$, and goal centerline at $\{0, 0, 0\}$, and the target being in the upper left corner of the goal $\boldsymbol{T} = \{0, -3, 2.3\}$. We also inserted a wall located at $\{5, 7.5, 0\}$ that was 0.5 m thick, 4 m wide and 2 m tall.

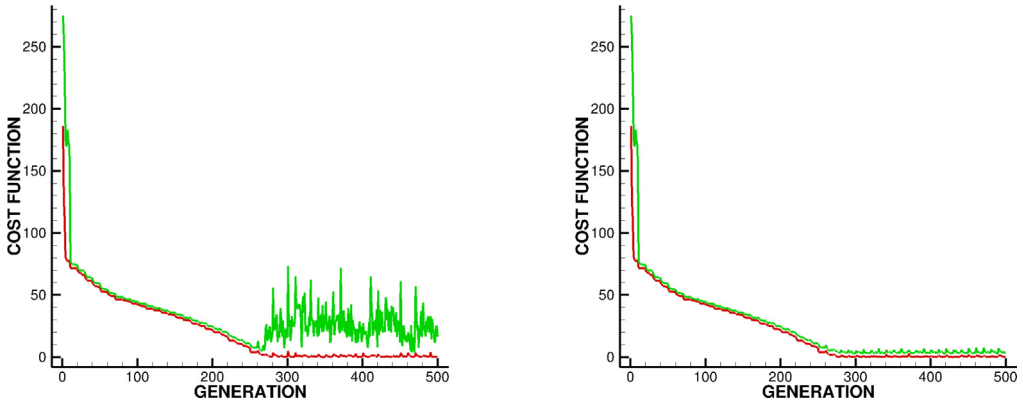

Fig. 8. Left: The reduction of the cost function for the 8 parameter set with the wall present. Shown are the best performing gene (design parameter set, in *red*) as a function of successive generations, as well as the average performance of the entire population of genes (designs, in *green*). Allowing the MLA/GA to readapt every 10 generations, leading to the nonmonotone reduction of the cost function. The cost function was reduced to $\Pi = 0.1158$, which implies, with ball radius of $R = 0.11$ m, a distance of $\|\boldsymbol{r} - \boldsymbol{T}\|$=0.1158 m, i,.e. *approximately a 1.27 cm difference between the ball center and the exact target. The optimal design is extremely tight after approximately 260 generations, and any deviation from the optimal leads to the wall stopping the ball (as exhibited by the green population average). Right: The reduction of the cost function for the 8 parameter set without the wall present. The optimal design is not extremely tight after approximately 260 generations, relative to the simulation with the wall.*.

One can interpret the term $\boldsymbol{\omega} \times \boldsymbol{r}_{cm \to p}$ as "velocity compensation" to induce the necessary spin, in addition to the needed center of mass velocity. Depending on the choice of $\boldsymbol{r}_p$, the compensation will change. It is important to recognize that this level of analysis is the effective result of player–ball contact. In order to provide more precise instructions to a player, more details on the complex impact event, accounting of the large deformability of the ball and players foot at the point of contact would need to be computed, involving Finite Element Methods, such those found in the well-known books of Wriggers [46,47].

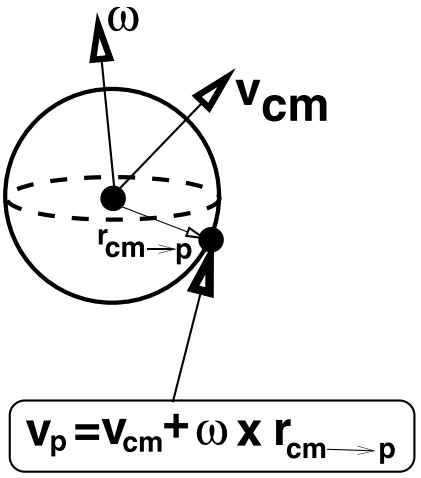

Fig. 9. The general relationship between the velocity of the mass center, desired spin and the point of contact. One can interpret the term $\boldsymbol{\omega} \times \boldsymbol{r}_{cm \to p}$ as "velocity compensation" to induce the necessary spin. Depending on the choice of $\boldsymbol{r}_p$, the compensation will change.

5. Conclusions

The objective of this work was to provide a framework to ascertain the optimal kick velocity, direction and spin that a player should impart to precisely hit a target goal with a bending kick, using machine-learning optimization. Specifically, the work developed a model consisting of a kicking player, resulting in a high-velocity spinning soccer ball, interacting with the surrounding fluid, subject to drag forces and the Magnus effect, which are both functions of the Reynolds number. The framework is designed to enable digital-twin type technologies, i.e. digital replicas that run in real time with the physical system on laptops or other mobile systems for rapid feedback. The overall guiding motivation is to provide a useful tool to rapidly train players. Numerical examples were provided to illustrate the process. As alluded to earlier, more detailed analysis of the solid–fluid interaction between the ball and surrounding air requires high-fidelity spatio-temporal models. For example, in Zohdi [48,49], more detailed, computationally intensive models were developed to characterize the motion of large-numbers of small-scale spheres embedded in a flowing fluid, where the dynamics of the spheres affects the dynamics of the fluid and vice-versa. In such a framework, a fully implicit Finite-Difference discretization of the Navier–Stokes equations was used for the fluid and a direct sphere-dynamics discretization is performed for the spheres. Because of the large computational difficulty and expense of a conforming spatial discretization needed for large numbers of embedded spheres, simplifying assumptions were made for the coupling, based on semi-analytical computation of drag-coefficients, which allows for the use of coarser meshes. Even after the se simplifications, the sphere–fluid system was strongly-coupled. The approach taken in that work was to construct a sub-model for each primary physical process. In order to resolve the coupling, a recursive staggering scheme was constructed, which was built on works found in Zohdi [48,49]. The procedure was as follows (at a given time increment): (1) each submodel equation (fluid or sphere-system) is solved individually, "freezing" the other (coupled) fields in the system, allowing only the primary field to be active, (2) after the solution of each submodel, the associated field variable was updated, and the next submodel was solved and (3) the process is then repeated, until convergence. The time-steps were adjusted to control the rates of convergence, which is dictated by changes in the overall physics. Specifically, the approach was a staggered implicit time-stepping scheme, with an internal recursion that automatically adapted the time-step sizes to control the rates of convergence within a time-step. If the process did not converge (below an error tolerance) within a preset number of iterations, the time-step was adapted (reduced) by utilizing an estimate of the spectral radius of the coupled system. The developed approach can be incorporated within any standard computational fluid mechanics code based on finite difference, finite element, finite volume or discrete/sphere element discretization (see Onate et al. [50–52] and Avci and Wriggers [53]). However, while useful in many industrial applications where high precision is required, the use of such a paradigm, for the relatively coarser application of interest in this work, appears to be unwarranted.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Youtube, Free kick of R. Carlos, 1997, https://www.youtube.com/watch?v=mZbVuoCFkVM.
[2] J.W.M. Bush, The aerodynamics of the beautiful game, 2013, in: C. Clanet (Ed.), Sports Physics, 2013, pp. 171–192, Les Editions de L'Ecole Polytechnique. Version: ISBN: 978-2-7302-1615-9.
[3] Wolfgang Demtroder, Experimentalphysik 1 Mechanik Und Warme, Springer-Verlag GmbH, Berlin, ISBN: 978-3-662-62727-3, 2021, p. 250, (9. Auflage 2021 ed.). OCLC 1222206116. Magnus Effekt einer Kugel unter idealisierten Bedingungen. www.physikerboard.de.
[4] Lyman Briggs, Effect of spin and speed on the lateral deflection (curve) of a baseball and the magnus effect for smooth spheres (PDF), Amer. J. Phys. 27 (8) (1959) 589–596, http://dx.doi.org/10.1119/1.1934921, Bibcode:1959AmJPh..27..589B.
[5] Milton Van Dyke, An album of fluid motion, in: Cross, Rod. Wind Tunnel Photographs(PDF), Stanford University, Physics Department, University of Sydney, 1982, p. 4.
[6] G. Magnus, Uber die Abweichung der Geschosse, Abhandlungen der Koniglichen Akademie der Wissenschaften zu Berlin, 1852, pp. 1–23.
[7] G. Magnus, Uber die Abweichung der Geschosse, Und: Uber Eine Abfallende Erscheinung Bei Rotierenden Korpern (on the Deviation of Projectiles, and: On A Sinking Phenomenon Among Rotating Bodies), Ann. Phys. 164 (1) (1853) 1–29.
[8] Lord Rayleigh, On the irregular flight of a tennis ball, in: Messenger of Mathematics, Vol. 7, 1877, pp. 14–16, Spin Axis. Trackman Golf. 17 2015.
[9] L.J. Clancy, Aerodynamics, Pitman Publishing Limited, London, ISBN: 0-273-01120-0, 1975.
[10] R.D. Mehta, Malinga's unique swing, in: The Wisden Cricketer, vol. 4, No. 10, Pitman Publishing Limited, 2007, p. 23.
[11] Magnus Effect. Wikipedia https://en.wikipedia.org/wiki/Magnus_effect.
[12] C.Y. Chow, An Introduction to Computational Fluid Dynamics, Wiley, New York, 1980.
[13] G. Plimpton, Fireworks: A History and Celebration, Doubleday, ISBN: 0-385-15414-3, 1984.
[14] A. St. Hill Brock, A History of Fireworks, George G. Harrap and Co, 1949.
[15] M.S. Russell, The chemistry of fireworks, in: Royal Society of Chemistry, Great Britain, ISBN: 978-0-85404-127-5, 2008.
[16] T. Shimizu, Fireworks: The Art, Science, and Technique, Pyrotechnica Publications, ISBN: 978-0-929388-05-2, 1996.
[17] S. Werrett, Fireworks: Pyrotechnic Arts and Sciences in European History, University of Chicago Press, ISBN: 978-0-226-89377-8, 2010.
[18] S. Kazuma, Hanabi no Hon, in: Fireworks Book. Tankosha, ISBN: 4-473-03177-2, 2004.
[19] S. Kazuma, Wonder of fireworks, in: Soft Bank Creative, ISBN: 978-4-7973-6450-7, 2011.
[20] V.K. Wingerden, I. Hesby, R. Eckhoff, Ignition of dust layers by mechanical sparks, in: Proceedings of 7th Global Congress on Process Safety, Chicago, Ill, 2011, 2011.
[21] A.C. Fernandez-Pello, Wildland fire spot ignition by sparks and firebrands, Fire Saf. J. 91 (2017) 2–10.
[22] G.E. Pleasance, J.A. Hart, An examination of particles from conductors clashing as possible source of bushfire ignition, in: State Electricity Commission of Victoria (SEC), Victoria, Australia, Research and Development Department, Report FM-1, 1977, 1977.
[23] A.D. Stokes, Fire ignition by copper particles of controlled size, J. Electr. Electron. Eng. (1990) 188–194.
[24] G. Rowntree, A. Stokes, Fire ignition of aluminum particles of controlled size, J. Electr. Electron. Eng. (1994) (1994) 117–123.
[25] R. Hadden, S. Scott, C. Lautenberger, C.A. Fernandez-Pello, Ignition of combustible fuel beds by hot particles: an experimental and theoretical study, Fire Technol. 47 (2011) 341–355.
[26] J.L. Urban, C.D. Zak, J. Song, A.C. Fernandez-Pello, Smolder spot ignition of natural fuels by a hot metal particle, Proc. Combust. Inst. (ISSN: 1540-7489) 36 (2) (2017) 3211–3218, http://dx.doi.org/10.1016/j.proci.2016.09.014.
[27] T.I. Zohdi, A note on firework blasts and qualitative parameter dependency, in: Proceedings of the Royal Society, 2016, http://dx.doi.org/10.1098/rspa.2015.0720.
[28] T.I. Zohdi, J. Cabalo, On the thermomechanics and footprint of fragmenting blasts, Internat. J. Engrg. Sci. 118 (2017) 28–39.
[29] T.I. Zohdi, Modeling the spatio-thermal fire hazard distribution of incandescent material ejecta in manufacturing, Comput. Mech. (2018) http://dx.doi.org/10.1007/s00466-018-1617-2.
[30] T.I. Zohdi, A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments, Comput. Methods Appl. Mech. Engrg. (2020) http://dx.doi.org/10.1016/j.cma.2020.112907.
[31] Schlichtling H., Boundary-layer theory, seventh ed., McGraw-Hill, New York, 1979.
[32] T.I. Zohdi, Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning, Comput. Methods Appl. Mech. Engrg. 331 (1) (2018) 343–362.
[33] T.I. Zohdi, The Game of Drones: rapid agent-based machine-learning models for multi-UAV path planning, Comput. Mech. (2019) http://dx.doi.org/10.1007/s00466-019-01761-9.

[34] Zohdi. T.I., A digital-twin framework for machine learning optimization of aerial fire fighting and pilot safety, Comput. Methods Appl. Mech. Engrg. 373 (1) (2021) 113446.
[35] T.I. Zohdi, A digital-twin and machine-learning framework for ventilation system optimization for capturing infectious disease respiratory emissions, Arch. Comput. Methods Eng. (2021) http://dx.doi.org/10.1007/s11831-021-09609-3.
[36] T.I. Zohdi, A digital-twin and machine-learning framework for the design of multiobjective agrophotovoltaic solar farms, Comput. Mech. (2021) http://dx.doi.org/10.1007/s00466-021-02035-z.
[37] T.I. Zohdi, A digital-twin and machine-learning framework for precise heat and energy management of data-centers, Comput. Mech. (2022) http://dx.doi.org/10.1007/s00466-022-02152-3.
[38] J.H. Holland, Adaptation in natural & artificial systems, in: Ann Arbor, Mich, University of Michigan Press, 1975.
[39] J.H. Holland, J.H. Miller, Artificial adaptive agents in economic theory (PDF), Amer. Econ. Rev. 81 (2) (1991) 365–371.
[40] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.
[41] L. Davis, Handbook of Genetic Algorithms, Thompson Computer Press, 1991.
[42] C. Onwubiko, Introduction to Engineering Design Optimization, Prentice Hall, 2000.
[43] D.E. Goldberg, K. Deb, Special issue on genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 121–124.
[44] D. Luenberger, Introduction to Linear & Nonlinear Programming, Addison-Wesley, Menlo Park, 1974.
[45] P. Gill, W. Murray, M. Wright, Practical Optimization, Academic Press, 1995.
[46] P. Wriggers, Computational contact mechanics, John-Wiley, 2002.
[47] P. Wriggers, Nonlinear Finite Element Analysis, Springer, 2008.
[48] T.I. Zohdi, Computation of strongly coupled multifield interaction in particle-fluid systems, Comput. Methods Appl. Mech. Engrg. 196 (2007) 3927–3950.
[49] T.I. Zohdi, Embedded electromagnetically sensitive particle motion in functionalized fluids, Comput. Part. Mech. 1 (2014) 27–45.
[50] E. Onate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows, Comput. Methods Appl. Mech. Engrg. 197 (19–20) (2008) 1777–1800.
[51] E. Onate, M.A. Celigueta, S.R. Idelsohn, F. Salazar, B. Suarez, Possibilities of the Particle Finite Element Method for fluid-soil–structure interaction problems, Comput. Mech. 48 (2011) 307–318.
[52] E. Onate, M.A. Celigueta, S. Latorre, G. Casas, R. Rossi, J. Rojek, Lagrangian analysis of multiscale particulate flows with the particle finite element method, Comput. Part. Mech. 1 (1) (2014) 85–102.
[53] B. Avci, P. Wriggers, A DEM-FEM coupling approach for the direct numerical simulation of 3D particulate flows, J. Appl. Mech. 79 (2012) 010901-(1-7).