Pattern Anal Applic
DOI 10.1007/s10044-012-0276-8

INDUSTRIAL AND COMMERCIAL APPLICATION

A methodology for detection and estimation in the analysis of golf putting

Micael S. Couceiro · David Portugal · Nuno Gonçalves · Rui Rocha · J. Miguel A. Luz · Carlos M. Figueiredo · Gonçalo Dias

Received: 12 October 2010 / Accepted: 22 May 2012

Abstract This paper presents a methodology for visual detection and parameter estimation to analyze the effects of the variability in the performance of golf putting. A digital camera was used in each trial to track the putt gesture. The detection of the horizontal position of the golf club was performed using a computer vision technique, followed by an estimation algorithm divided in two different stages. On a first stage, diverse nonlinear estimation techniques were used and evaluated to extract a sinusoidal model of each trial. Secondly, several expert golf player trials were analyzed and, based on the results of the first stage, the Darwinian particle swarm optimization (DPSO) technique was employed to obtain a complete kinematical analysis and a characterization of each player's putting technique. In this work, it is intended not only to test the performance of the DPSO method, but also to present a novel study in this field by identifying a putting "signature" of each player.

Keywords Golf putting · Motion analysis · Detection · Estimation · Signature

M. S. Couceiro (✉) · D. Portugal · N. Gonçalves · R. Rocha
Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal
e-mail: micaelcouceiro@isr.uc.pt

D. Portugal
e-mail: davidbsp@isr.uc.pt

N. Gonçalves
e-mail: nunogon@isr.uc.pt

R. Rocha
e-mail: rprocha@isr.uc.pt

M. S. Couceiro · J. M. A. Luz · C. M. Figueiredo
RoboCorp, Electrotechnics Engineering Department, Engineering Institute of Coimbra, Rua Pedro Nunes, 3031-601 Coimbra, Portugal
e-mail: miguel.luz@isec.pt

C. M. Figueiredo
e-mail: cfigueiredo@isec.pt

G. Dias
RoboCorp, Faculty of Sport Sciences and Physical Education, University of Coimbra, University Stadium of Coimbra, 3040-156 Coimbra, Portugal
e-mail: goncalodias@fcdef.uc.pt

1 Introduction

According to the Merriam-Webster's Dictionary [1], the putting technique, or simply the "putt", is defined as a light golf stroke made on the putting green in an effort to place the ball into the hole. Hence, the putt is used in short distance shots on or near the green, as seen in Fig. 1. Similarly, "putter" may refer to a golf club used in the putting stroke or the player who is attempting to putt. In this work, the term "putter" is used solely in the sense of a golf club.

Recent researches concerning putt execution were carried out [2–6]. The biomechanical aspects of this gesture were addressed in some other studies [7–10], and the majority of the known research about this subject were made in laboratory context, i.e., indoor [3, 4, 11–13].

However, only a few studies analyzing process variables, such as the position, velocity or acceleration in the golf club during putt execution (linear or angular) [7–9] have been made and researches using automatic tracking of the ball's trajectory and movement on the green or in laboratory are not known.

Attending to the information exposed, this work presents the experimental design and methodological aspects in the analysis of the effects of variability in the golf putting performance of expert subjects, using an algorithm that automatically

Published online: 08 June 2012

Fig. 1 Putter movement action parameter analysis: **a** initial stage, **b** back swing, **c** down swing and ball impact, **d** follow-through

detects stationary and dynamic objects (e.g., in golf's putt execution). This algorithm is used to obtain a point cloud of the horizontal position of the putter during the execution. Consequently, many process variables, such as the trajectory function, can be obtained by estimation of a sinusoidal model to fit to the cloud points by means of different estimation techniques. In a first stage, many estimation techniques with different performance are analyzed. In a second stage, one of these techniques is chosen to estimate the trajectory model, by computing the parameters of a function composed by a sum of three sinusoids. Finally, an individual study of each expert player's putting trial is conducted to identify relationships between different executions of the same player, thus extracting putting "signatures" for every player.

In the next section, the experimental design of the experiences is presented. After which, the detection and the main estimation algorithms are described. Subsequently, the experimental results of the mentioned stages are exposed and discussed. Finally, the article ends with conclusion and future work.

2 Related work

As mentioned before, this study can be divided in two distinct steps. Firstly, the dynamic position of the golf club is detected, by a computer vision technique, during the execution of the putt, and a model of its trajectory is estimated using different existing methods to analyze their performance. Secondly, the estimation method with the best performance in this preliminary phase is used to carry out a study of the relationship between different trials of the same player, by analysis of several experiments using different subjects, to identify a putting "signature" of each player. To accomplish this goal, it is important to establish a detection algorithm and an estimation algorithm.

2.1 Detection

The problem of detection and object tracking has been subject to numerous studies and has gained considerable interest in many research fields, such as biological motion [14], human vision systems [15], traffic monitoring [16, 17], pedestrian protection systems [18, 19] or surveillance [20].

The methods for object tracking can be subdivided into two main groups: deterministic [21–24] and probabilistic [25], within which the Bayesian techniques are the most popular.

Within the group of deterministic methods, the mean-shift algorithm is one of the most widely used. The mean-shift algorithm is a gradient-based iterative technique originally proposed in Ref. [26] and uses different kernels, such as Gaussian or Epanechnikov for representing a probability density function, moving to a kernel-weighted average of the observations within a smoothing window. This computation is repeated until convergence is attained at a local density mode. It was further extended to computer vision problems [21, 23, 24]. In Ref. [24], the similarity between the target region and the target candidates in the next video frame is evaluated using a metric based on the Bhattacharyya coefficient. Based on the mean-shift vector, received as an estimation of the gradient of the Bhattacharyya function, the new object state estimate is calculated.

The mean-shift algorithm has been combined with particle filtering techniques, and as a result kernel particle filters [27] and hybrid particle filters [28] were proposed combining the advantages of both approaches. Particles are moved into more likely regions and hence the performance of these hybrid particle filters is significantly improved. Other related hybrid particle filters combined with the mean shift are proposed [29–32]. The accuracy of the mean-shift techniques depend on the chosen kernel and the number of iterations in the gradient process. A drawback of this algorithm is that sometimes local optima are found instead of the global one.

Most video object tracking techniques are region based, which means that the object of interest is contained within a region, often of a rectangular or circular shape. This region is then tracked in a sequence of video frames based on certain features, such as color, texture, edges, shape and/or their combinations [25, 33, 34].

There have been several works in literature exploring pixel-based color detection methods for recognition or tracking of features in images. Very frequently, these methods are used together with region detection (or image segmentation). Most authors claim that color allows fast processing and is robust to geometric variations; therefore, using such techniques is advantageous. They can be applied in many fields like robotics, for example on tracking systems for soccer-player robots. In Ref. [35], the vision system described models the ball by its position, size and color in HSI space. Firstly, saturation and intensity masks are applied in the search window. Then, the final segmentation decision is done using hue distance to the model's color in the filtered pixels. Also a backup segmentation was developed using the RGB color space and analyzing pixel distances. A dynamic adjustment algorithm of the search window size was also implemented based on predictions of the objects' position. Larger window sizes are only necessary when objects move very fast. Generally, there is no need to process most parts of the image, which is also the case in the present work, as it will be shown later on.

Another vision system implemented for assistance in agriculture [36] uses color and shape analysis on near-ground image of cereal crops in order to detect weeds. The pixels in the images are discriminated by means of color analysis and applying thresholds. The main application of such system is to monitor weeds and spraying herbicide simultaneously in an efficient spatially variable manner.

Several studies on detection of human features also use these techniques. Vezhnevets [37] compares published pixel-based skin detection methods by summarizing their advantages, disadvantages and characteristic features. In these methods, each pixel is classified as skin or non-skin individually independently from its neighbors, and human skin color is used for face detection. Also, a study [38] uses skin-hue classification to identify and track likely body parts within the silhouette of a user on a real-time person tracking and recognition system in crowded or unknown environments, combining stereo vision, color and face detection modules.

In Ref. [39] an interesting approach is taken, as the method uses a three-layered data association scheme with graph-theory formulation to track tennis balls, in broadcasting video sequences. Several works can be found in the literature on using graph theory to solve the data association problem [40, 41]. Each object candidate is usually modeled as a node in a graph and the object trajectory is identified by looking for the optimal path. In their method, the association problem is mapped onto a graph and solved using an improved all-pairs shortest path formulation and path level analysis, resulting in a fully automatic data association algorithm able to handle multiple object scenarios. The proposed data association algorithm has been applied on large data sets containing broadcasted tennis sequences from three tournaments. Comparative experiments show its robustness, as good results are obtained on sequences where other data association methods perform poorly or fail completely.

This last algorithm presents an alternative way to solve tracking problems, as those are most commonly solved with estimation techniques. A review on this matter is given in the next section.

2.2 Estimation

The problem of tracking dynamic objects and estimating their time-varying position has been studied extensively in robotics, engineering, computer vision and several other fields [42]. The problem is hard because the appearance of objects is ambiguous, partly occluded, may vary quickly over time and is perceived via a high-dimensional measurement space.

Five different estimation techniques were studied, applied and compared in this work. One popular first-order optimization algorithm used in many engineering related works is the Gradient Descent. In this method, the search is carried through proportional steps in the direction of the negative of the gradient, or the approximate gradient, of the function at the current point to find a local minimum. It has been applied in the literature, for example, for face alignment in computer recognition systems [43].

Another vastly used method is Pattern Search, which is a similar approach to the Gradient Descent. However, it does not compute the gradient, meaning that it can be used with non-differentiable functions. In this case, a descent search direction is produced, by varying the parameters of the problem with different step sizes, aiming to obtain a fit to the experimental data. It has been successfully applied in model selection of support vector machines [44] and for transformation function search in automatic image registration [45] among others. Both the Gradient Descent and the Pattern Search methods are more suitable for low-dimensional optimization problems.

In addition, the simplex method was also studied. This is an optimization method to numerically solve linear programming problems by searching optimal solutions in the vertices of the admissible region of the space, considering all constraints, and iteratively improving the objective function. It is perhaps the most popular optimization algorithm for linear problems with low dimensions, being applied previously in contexts like particle accelerator control [46].

A nonlinear heuristic version of the simplex method called the downhill simplex or Nelder–Mead algorithm was utilized. This algorithm is more suitable for minimizing objective functions in a many-dimensional space, such as in this work. Among previous works, Ref. [47] used a modified version of the Nelder–Mead algorithm to minimize an objective function related to the displacement of the joints of a robotic manipulator by visual servoing.

All the referred methods proved to be time consuming in problems with multiple dimensions and relaxed restrictions, as it will be clear later on. As a consequence, the focus was shifted to an alternative approach: the particle swarm optimization (PSO) because of its recent popularity and performance obtained in various studies such as robotics [48–50], electrical systems [51] and sport sciences [52].

A general problem with PSO and other optimization heuristics is becoming trapped in a local optimum. PSO may perform well on a given problem, but may fail to do so in problems with different characteristics. In order to overcome this issue, many authors have suggested adjustments to the PSO algorithm's parameters, combining, for example, fuzzy logic (FAPSO), where the inertia weight is dynamically adjusted using fuzzy "IF–THEN" rules [53], or Gaussian approaches (GPSO) where the inertia constant is no longer needed and the acceleration constants are replaced by random numbers with Gaussian distributions [42].

More recently, Pires et al. [54] used fractional calculus to control the convergence rate of the PSO. The authors rearrange the original velocity equation (1) to modify the order of the velocity derivative.

Many authors have considered incorporating selection, mutation and crossover, as well as differential evolution (DE), into the PSO algorithm. The main goal is to increase the diversity of the population by preventing the particles from moving too close to each other and collide [55, 56] or to enable self-adaptation of parameters such as the constriction factor, acceleration constants [57] or inertia weight [58].

The fusion between genetic algorithms (GA) and the PSO led to the GA–PSO [59]. GA–PSO combines the advantages of swarm intelligence and natural selection mechanisms, to increase the number of highly evaluated agents, while decreasing the number of lowly evaluated agents at each iteration step.

Similar to the last one, the EPSO is an evolutionary approach that incorporates a selection procedure into the original PSO algorithm, as well as self-adapting properties for its parameters. This algorithm makes use of a tournament selection method commonly applied in evolutionary programming (EP) [60]. Based on the EPSO, a differential evolution operator has been proposed to improve the performance of the algorithm in two different ways. The first one [61] applies the differential evolution operator to the particle's best position to eliminate the particles falling into local minima (DEPSO), while the second one [62] applies it to find the optimal parameters (inertia and acceleration constants) for the canonical PSO (C-PSO).

One of the common drawbacks in most of the PSO variations is the significant increase in the computational effort. Having that in mind, an evolutionary method based on the PSO, called Darwinian particle swarm optimization (DPSO), was implemented and tested. This method was chosen due to its potential to escape local optima, resulting into high-quality performance, as shown in [63]. Additionally, it is a recently proposed technique suffering from an apparent lack of application in engineering problems and, at the same time, requires less computational effort when compared with the previously described PSO variants. In Sect. 5, both PSO and DPSO are reviewed in detail.

3 Experimental design

The experimental design and methodological aspects that support this research are herein presented. The adopted task was the golf putt, implying the strike of a ball (Titleist; Model Pro V1) with a putter (Putter Jumbo Black Beauty; size 35; standard) on a horizontal and still surface, placed on the ground over a ramp.

3.1 Apparatus and procedure

The apparatus included an artificial, rectangular, green and plain carpet with no flaws, which is commonly used by Minigolf professionals and is quite similar to the green natural surface texture; it was 10-m long, 2-m wide and 4-mm thick.

The ball's rolling speed on the carpet was measured with a stimpmeter, corresponding to 10 m/s, which is an acceptable value accordingly to the green's validation criteria of the Professional Golf Association (PGA Tour).

A real golf hole was placed 3.5 m away from the carpet's edge and 1 m away from each lateral extremity. A white dot marked the putting location at 2 m. The dot is in the same direction of the hole and 1 m of each lateral extremity of the carpet. Under the carpet, a 1-m long ramp was placed, leveling the carpet's surface, 10-cm high.

On the top side of the ramp there was a platform 4-m long (left side of Fig. 2). The ramp allows the ball to travel from the lower level up to the hole.

The computational tool MatLab was used to deal and analyze all the data. Using this application, one can perform statistic analysis as well as computing mathematical models based on cinematic and kinetics of real models [64].

3.2 Data recording

To perform this study, a digital Casio Exilim/High Speed EX-FH25 camera was used. The autonomy of the digital camera was also considered and, in order to smoothly record the entire session without interruptions, rechargeable 2,700-mA batteries and 16-GB memory cards were used.

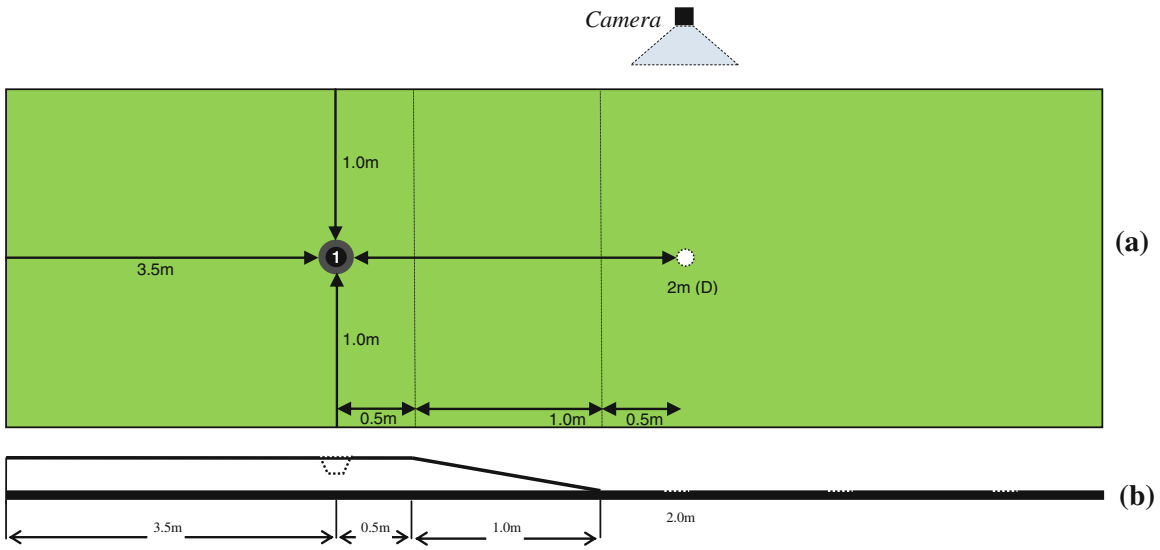

Fig. 2 Experimental device: **a** *top view*, **b** *side view*

The digital camera was placed 55 cm above the ground, heading forward and 4 m away from the experimental device, in front of the subject. These procedures were based on Knudson and Morrison's work [65], which suggests shooting distances of 2–10 m in similar studies.

The camera worked on its tripod and all the positioning and calibration features mentioned were used in the same way for the entire study. It shot at 210 fps (frames per second) with a resolution of 480 × 360 pixels and a focal length of 26 mm. The experimental device and digital camera were always in the exactly same place, so that everything was recorded under the same conditions, guaranteeing reliability for later data analysis.

Several previous studies on putting performance analysis used digital cameras recording from 25 fps up to 50 fps [66]. This confirms that 210 fps is an adequate frame rate to study a gesture as precise as the golf putt.

Logically, to be able to capture the scene with such a high frame rate in real time, the image quality must be worsened as both the image sensor exposure time and the resolution are reduced, hence the 480 × 360 resolution, which affects detections in the images and parameters estimation. This represents an interesting and realistic challenge due to noise in the captured images.

Digital camera recordings provide information about the golf putting action parameters in distinct stages (Fig. 1) to estimate the trajectory of the putter during the movement.

4 Detection algorithm

In this study, as a controlled environment was created, the following detection algorithm was used to detect the putter's head, through the red marker, according to the RGB range values defined (Fig. 3b):

1. Select the region of interest (Fig. 3b), within the entire frame (Fig. 3a), to be analyzed.
2. Analyze the region of interest of the current frame searching for pixels with RGB value within the defined RGB range.
3. If a pixel under condition (2) is found, then verify if it is inside a blob with at least the considered area of 8 pixels within the same RGB range.

Analyzing the video will result in a vector that includes the object position in the corresponding frame (pixel/frame) (i.e., trajectory). Optionally, we can:

4. Convert the pixel/frame value of the object in m/s (ISU).

As these digital cameras' lenses provide a considerable depth of field, a reference in the same plane of the analyzed gesture is necessary to perform the conversion to m/s. This reference is the putt's metallic part length of 585 mm.

The algorithm presented above is computationally efficient; it relies on simple image processing techniques and ensures satisfactory results.

The chart presented in Fig. 4 shows an example of a point cloud that represents the detected position, in the horizontal plane, of a golf club during putting execution of an expert subject.

As it is clear in Fig. 4, the detection algorithm's output has some lacking data. This happens when the conditions of the second step of the algorithm are not met. In such cases, the detection is skipped in the corresponding time instant to avoid introduction of errors. In order to classify

Fig. 3 Example of a registered scene: **a** full frame, **b** region of interest with the defined range of RGB values for putter detection

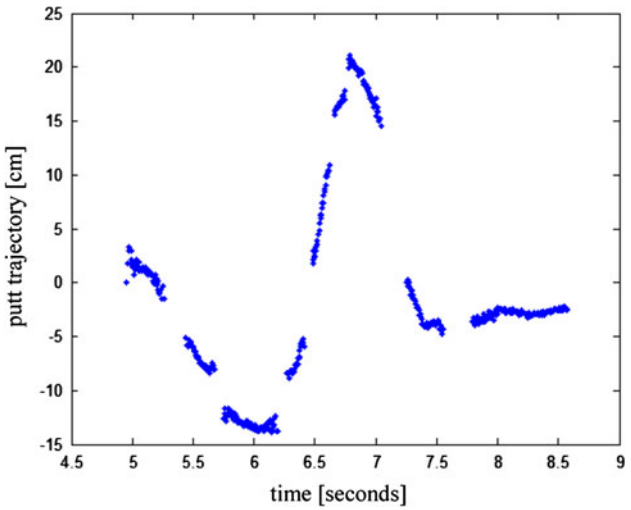

Fig. 4 Example of a point cloud obtained with the detection algorithm

the point cloud, linear and nonlinear estimation techniques were studied to fit the acquired points of the cloud to a sinusoidal function, thus obtaining a mathematical model to describe the putter's position during the execution of the play. In the next section, the PSO and DPSO estimation techniques are analyzed.

5 Estimation algorithms

In this section, a detailed review on the two most significant algorithms are presented in this work: in particular, the PSO and the DPSO.

5.1 Particle swarm optimization

Particle swarm optimization is a population-based stochastic optimization technique. In PSO, the candidate solutions are called particles. These particles travel through the search space to find an optimal solution, by interacting and sharing information with neighbor particles, namely their individual best solution (local best) and computing the neighborhood best. Also, in each step of the procedure, the global best solution obtained in the entire swarm is updated. Using all of this information, particles realize the locations of the search space where success was obtained and are guided by these successes.

In each step of the algorithm, a fitness function is used to evaluate the particle success. To model the swarm, each particle n moves in a multidimensional space according to position (x_n) and velocity (v_n) values, which are highly dependent on local best $(\breve{x}_n)$, neighborhood best $(\breve{n}_n)$ and global best $(\breve{g}_n)$ information:

$$v_n = wv_n + \rho_1 r_1(\breve{g}_n - x_n) + \rho_2 r_2(\breve{x}_n - x_n) + \rho_3 r_3(\breve{n}_n - x_n) \quad (1)$$

$$x_n = x_n + v_n \quad (2)$$

The coefficients w, ρ_1, ρ_2 and ρ_3 assign weights to the inertial influence, the global best, the local best and the neighborhood best when determining the new velocity, respectively. Typically, the inertial influence is set to a value slightly less than 1. ρ_1, ρ_2 and ρ_3 are constant integer values, which represent "cognitive" and "social" components. However, different results can be obtained by assigning different influences for each component. For example, several works do not consider the neighborhood best and ρ_3 is set to zero. Depending on the application and the characteristics of the problem, tuning these parameters properly will lead to better results. The parameters r_1, r_2 and r_3 are random vectors with each component generally a uniform random number between 0 and 1. The intent is to multiply a new random component per velocity dimension, rather than multiplying the same component with each particle's velocity dimension.

Algorithm 1. ***PSO*** **Algorithm**

```
Initialize swarm (Initialize x_n, v_n, x̆_n, n̆_n and ğ_n)
Loop:
    for all particles
            Evaluate the fitness of each particle
            Update x̆_n, n̆_n and ğ_n
            Update v_n and x_n
    end
until stopping criteria (convergence)
```

In the beginning, the particles' velocities are set to zero and their position is randomly set within the boundaries of the search space (Algorithm 1). The local, neighborhood and global bests are initialized with the worst possible values, taking into account the nature of the problem. There are other few parameters that need to be adjusted:

- Population size—very important to optimize to get overall good solutions in acceptable time.
- Stopping criteria—it can be a predefined number of iterations without getting better results or other criteria, depending on the problem.

Particle swarm optimization reveals an effect of implicit communication between particles (similar to broadcasting) by updating neighborhood and global information, which affects the velocity and consequent position of particles. Also, there is a stochastic exploration effect due to the introduction of the random multipliers (r_1, r_2 and r_3).

5.2 Darwinian particle swarm optimization

Many variations of the PSO algorithm have been presented since its first description in [67]. In this work, we have focused in an algorithm which incorporates techniques of evolutionary approaches to the PSO: the DPSO [63].

Despite the similarities between the PSO and genetic algorithms (GAs) such as randomly generated population, fitness function evaluation, population update, search for optimality with random techniques and not guaranteeing success, PSO does not use genetic operators like crossover and mutation, and thus is not considered an evolutionary technique. On the other hand, the DPSO extends the PSO to determine if natural selection (Darwinian principle of survival of the fittest) can enhance the ability of the PSO algorithm to escape from local optima. The idea is to run many simultaneous parallel PSO algorithms, each one a different swarm, on the same test problem and apply a simple selection mechanism. When a search tends to a local optimum, the search in that area is simply discarded and another area is searched instead.

In this approach, at each step, swarms that get better are rewarded (extend particle life or spawn a new descendent) and swarms which stagnate are punished (reduce swarm life or delete particles). To analyze the general state of each swarm, the fitness of all particles is evaluated and the neighborhood and individual best positions of each of the particles are updated. If a new global solution is found, a new particle is spawned. A particle is deleted if the swarm fails to find a fitter state in a defined number of steps.

Algorithm 2. ***DPSO*** **Algorithm**

Main Program Loop

```
For each swarm in the collection
    Evolve the swarm (Evolve Swarm Algorithm: right)
    Allow the swarm to spawn
    Delete "failed" swarms
```

Evolve Swarm Algorithm

```
For each particle in the swarm
        Update Particles' Fitness
        Update Particles' Best
        Move Particle
If swarm gets better
        Reward swarm: spawn particle:
        extend swarm life
If swarm has not improved
        Punish swarm: possibly delete par-
        ticle: reduce swarm life
```

Some simple rules are followed to delete a swarm, delete particles, and spawn a new swarm and a new particle:

- When the swarm population falls below a minimum bound, the swarm is deleted.
- The worst performing particle in the swarm is deleted when a maximum threshold number of steps (search counter SS_c^{max}) without improving the fitness function is reached. After the deletion of the particle, instead of being set to zero, the counter is reset to a value approaching the threshold number, according to:

$$SS_c(N_{kill}) = SS_c^{max} 1 - \frac{1}{N_{kill} + 1} \quad (3)$$

  with N_{kill} being the number of particles deleted from the swarm over a period in which there was no improvement in fitness.
- To spawn a new swarm, a swarm must not have any particle ever deleted and the maximum number of swarms must not be exceeded. Still, the new swarm is only created with a probability of $p = f/NS$, with f a random number in [0,1] and NS the number of swarms. This factor avoids the creation of newer swarms when a

large numbers of swarms exist. The parent swarm is unaffected and half of the parent's particles are selected at random for the child swarm and half of the particles of a random member of the swarm collection are also selected. If the swarm initial population number is not obtained, the rest of the particles are randomly initialized and added to the new swarm.
- A particle is spawned whenever a swarm achieves a new global best and the maximum defined population of a swarm has not been reached.

Like the PSO, a few parameters also need to be adjusted to run the algorithm efficiently:

- social and cognitive coefficients of v_n;
- initial swarm population;
- maximum and minimum swarm population;
- initial number of swarms;
- maximum and minimum number of swarms;
- stagnancy threshold;
- maximum and minimum velocity value.

Later on, the results obtained using both the PSO and DPSO will be shown and discussed.

6 Experimental results

By analysis of the shape of various point clouds given by the detection algorithm, it was clear that to model the putter's horizontal position in time, one ought to use a sinusoidal-like function.

Nevertheless, a function composed by only one sinusoid was not precise enough to describe the movement, as it is clear in f_1 of Fig. 5, which results, in this case, in a mean-squared error (MSE) of 2.6568 units. This happens because the amplitude, angular frequency and phase of the descending half-wave, which corresponds to the player's backswing and downswing, is usually different from the ascending half-wave, which corresponds to the ball's impact and follow-through. These disparities could not be represented using solely one sinusoid wave. Therefore, to obtain a more precise model a sum of sinusoid waves was employed. However, a compromise between precision and complexity of the problem had to be assumed, because each sinusoid added three more dimensions to the estimation problem (amplitude, angular frequency and phase of the corresponding sine wave). In order not to let the complexity of the problem grow inappropriately, a function composed of the sum of three sinusoids was used (f_3 of Fig. 5), due to its precision, with an MSE of 0.6926, as compared to using solely a sum of two sinusoids with an MSE of 0.7124 (f_2 of Fig. 5).

Thus, having the estimation function defined as a sum of three sine waves, each of the three parameters of each wave needs to be estimated, resulting in a nine-dimension estimation problem which attempts to minimize the mean-squared estimation error for every experiment, to obtain a precise function that describes the horizontal position of the golf club during putting execution.

6.1 First stage: preliminary results using different estimation algorithms

In this phase, a preliminary analysis of all five estimation methods mentioned in Sect. 3 was conducted to verify which one was better suited for the problem in hand. Three distinct trials of three different subjects were used in this first study. The goal was to check and compare the performance of each of the estimation techniques.

All algorithms were run under the same conditions, with the same restrictions and with random initialization of all variables. Below, a summary of those conditions is presented:

Model with 9 dimensions $[a_1\ b_1\ c_1\ a_2\ b_2\ c_2\ a_3\ b_3\ c_3]$:

Restrictions:

Amplitude	$a_1 \in [0, 35]$; $a_2 \in [0, 35]$; $a_3 \in [0, 35]$;
Angular Frequency	$b_1 \in [0, 0.03]$; $b_2, b_3 \in [0, 0.05]$;
Phase	$c_1, c_2, c_3 \in [-\pi, \pi]$;

Fitness Function:

Minimize the mean square error between the observed points and the estimated solution.

Stopping Criteria:

Threshold MSE = 0.75 or Maximum Running Time = 1,000 s.

Algorithms:

- Gradient Descent Interior-Point Basic
- Pattern Search Latin Hypercube
- Downhill Simplex
- *PSO* with 500 particles
- *DPSO* with 10 initial swarms and 20 initial particles

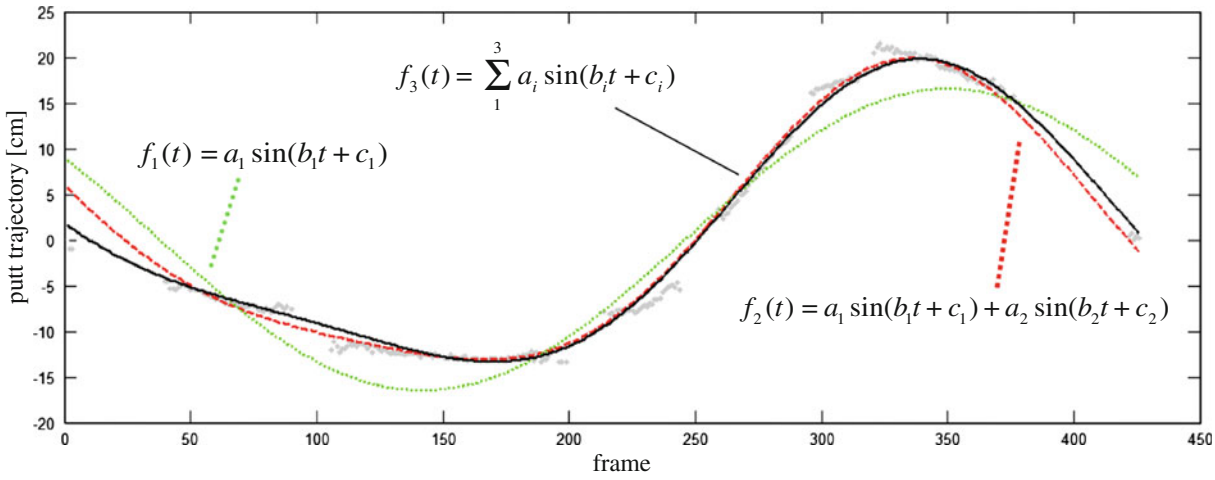

Fig. 5 Fitting sinusoidal functions to a point cloud, representing the position of a golf club during putting execution

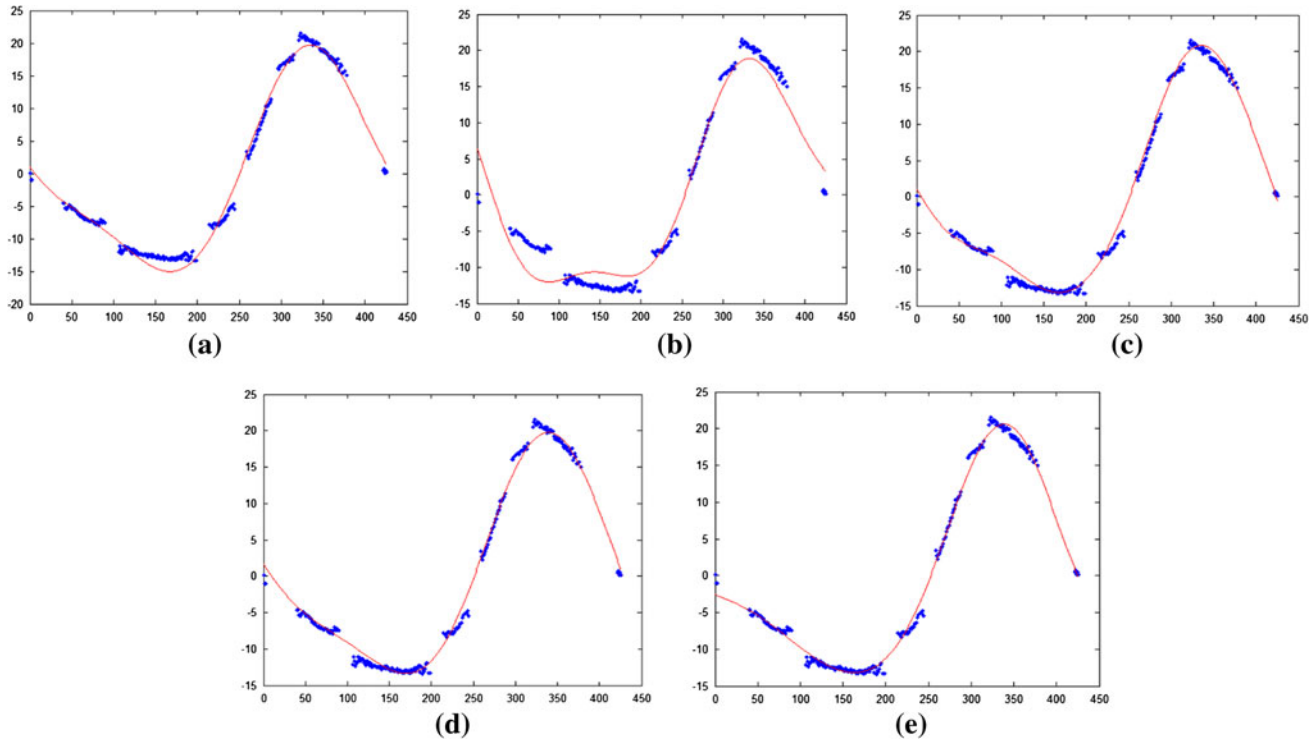

Fig. 6 Results for the first data set—frame versus putter trajectory (cm); **a** Gradient Descent, **b** Pattern Search, **c** Downhill Simplex, **d** PSO, **e** DPSO

Figures 6, 7 and 8 and Table 1 clearly show the superior performance obtained by the DPSO algorithm for all three cases, followed by PSO, which was only slightly inferior in the second data set when compared with pattern search and achieved second best results in the first and third data set. The other three algorithms generally obtained inferior results.

These results depict that population-based and evolutionary algorithms are more suited for complex multi-dimensional optimization problems. However, both the PSO and the DPSO need some predefined parameters, as opposed to gradient descent, pattern search and the downhill simplex. In these experiences, the following parameters were used:

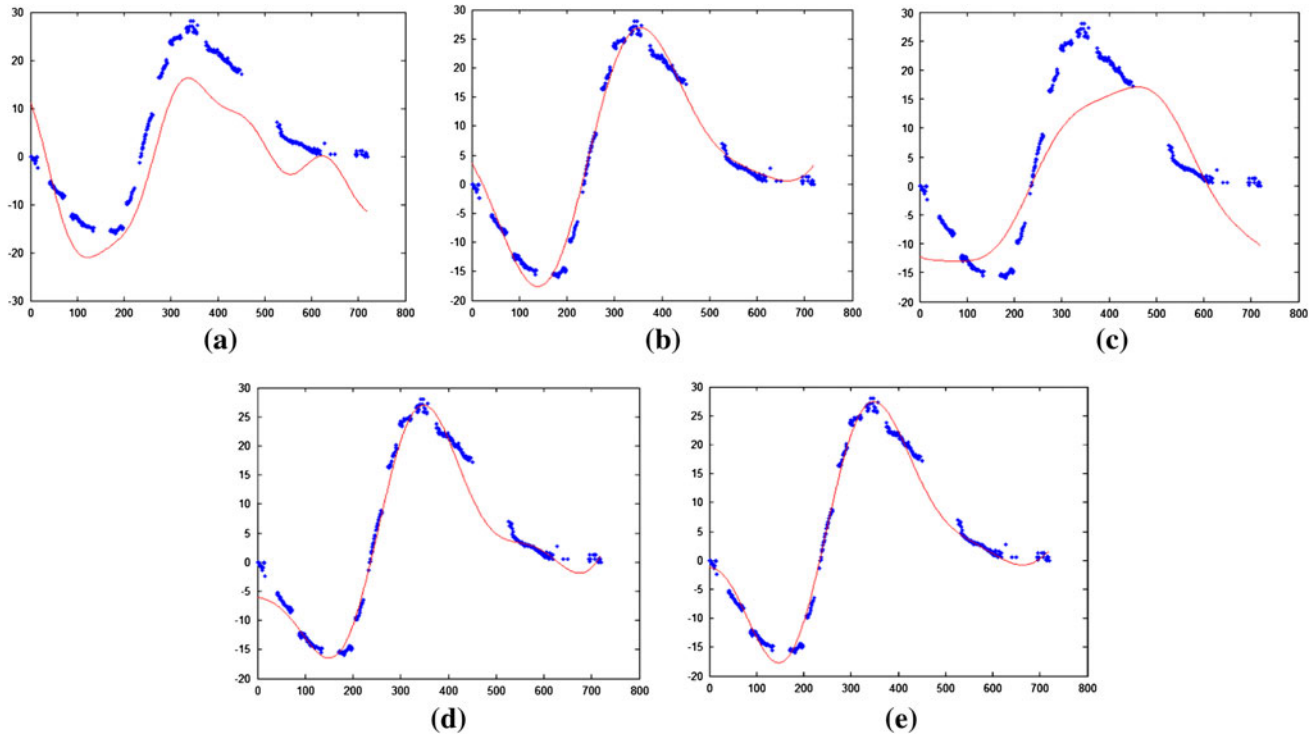

Fig. 7 Results for the second data set—frame versus putter trajectory (cm); **a** Gradient Descent, **b** Pattern Search, **c** Downhill Simplex, **d** PSO, **e** DPSO

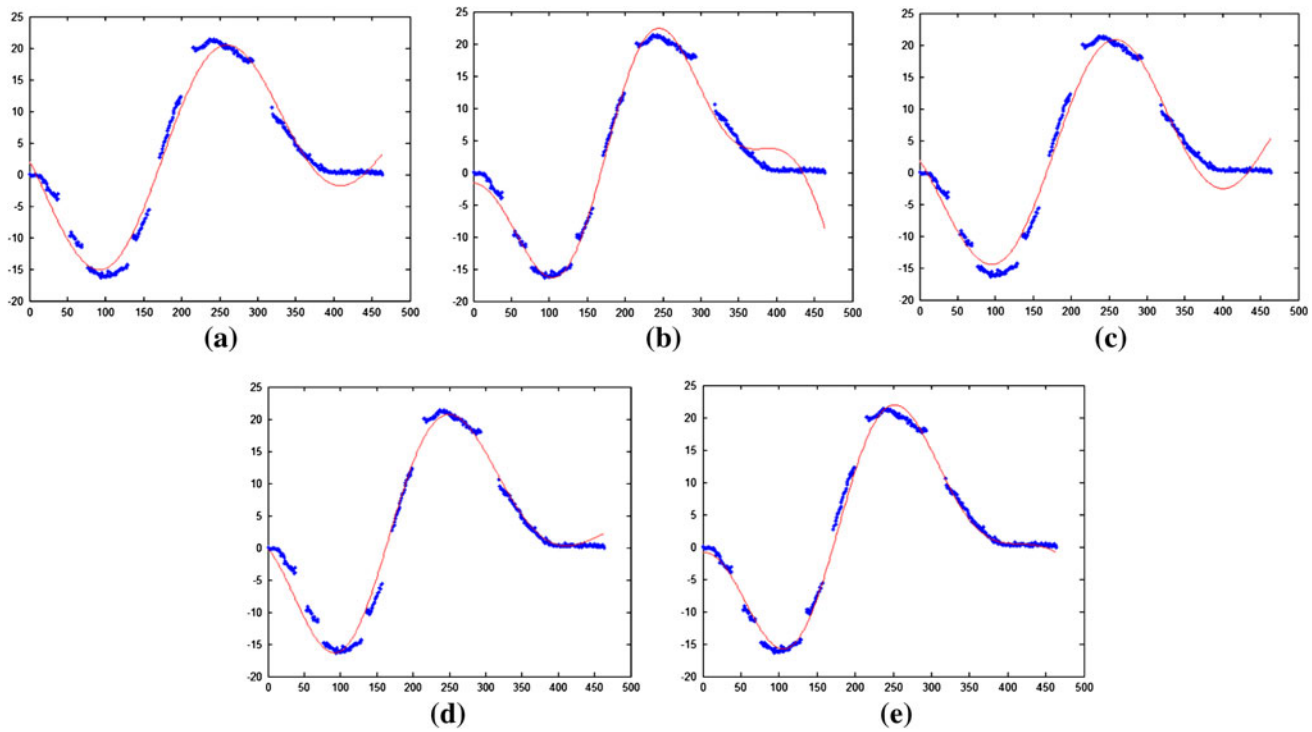

Fig. 8 Results for the third data set—frame versus putter trajectory (cm); **a** Gradient Descent, **b** Pattern Search, **c** Downhill Simplex, **d** PSO, **e** DPSO

Table 1 Comparative results for all three data s ets

	First data set			Second data set			Third data set		
	Iterations	Running time (s)	MSE	Iterations	Running time (s)	MSE	Iterations	Running time (s)	MSE
Gradient Descent	27	1,000	0.9270	47	1,000	7.2550	26	1,000	1.4989
Pattern Search	61	1,000	1.7989	71	1,000	1.4315	71	1,000	1.4087
Downhill Simplex	325	952	0.7147	471	1,000	5.7457	500	1,000	1.6770
PSO	3,463	312	0.6926	8,527	1,000	1.4391	8,754	1,000	0.9789
DPSO	87	102	0.5658	178	1,000	1.0001	43	8	0.7368

$$\text{PSO}: \quad \omega = 0,85; \quad \rho_1 = \rho_2 = 1; \quad \rho_3 = 0;$$

$$\text{DPSO}: \quad \omega = 0,6; \quad \rho_1 = \rho_2 = 0.5; \quad \rho_3 = 0;$$

Maximum particles per swarm : 100

Minimum particles per swarm : 15

Maximum number of swarms : 100

Minimum number of swarms : 1

Stagnancy threshold : 5

These parameters guaranteed the above fine results. Note that the neighborhood influence between particles was canceled ($\rho_3 = 0$) for both algorithms. Also, velocity limits of ± 0.5 were imposed in both cases to prevent explosion of particle velocities and each particle's position was verified in every iteration to guarantee that it moved inside the boundaries imposed by the constraints of all dimensions of the problem.

At this point, both PSO and DPSO algorithms have shown excellent search abilities for the considered data set but, as occurring with other algorithms, may lose their efficacy when applied to large and complex problems, e.g., problem instances with high dimensionality. There are several benchmark test functions commonly used to test the optimization algorithms' efficacy, one of those being the Rastrigin function [68]. This benchmark function has many, regularly distributed local optima, presenting itself as a challenge for optimization algorithms (Fig. 9) defined as:

$$\text{Ras}(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(\cos 2\pi x_i) + 10\right), \quad -5.12 < x_i < 5.12 \tag{4}$$

where D is the dimension of the problem and $x = (x_1, x_2, \ldots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix). It is typically considered a dimension of 30, i.e., $D = 30$, when considering benchmark functions, which was the dimension used in our evaluation.

In our case, test groups of 100 trials with 500 iterations each were considered for both PSO and DPSO with the previously defined parameters. To present the results, the median of the best solution of the 100 trials was taken as the final output (Fig. 10).

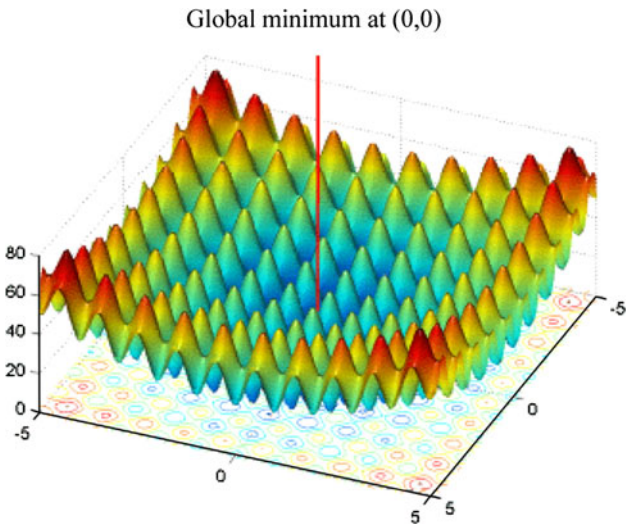

Fig. 9 Representation of Rastrigin function ($D = 2$)

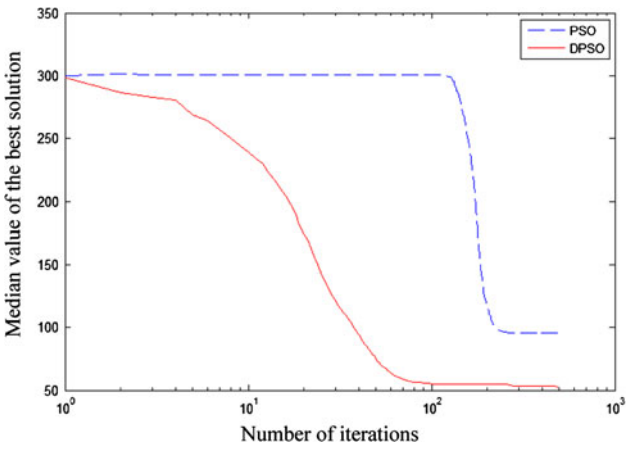

Fig. 10 Evolution of PSO and DPSO performance in the Rastrigin function

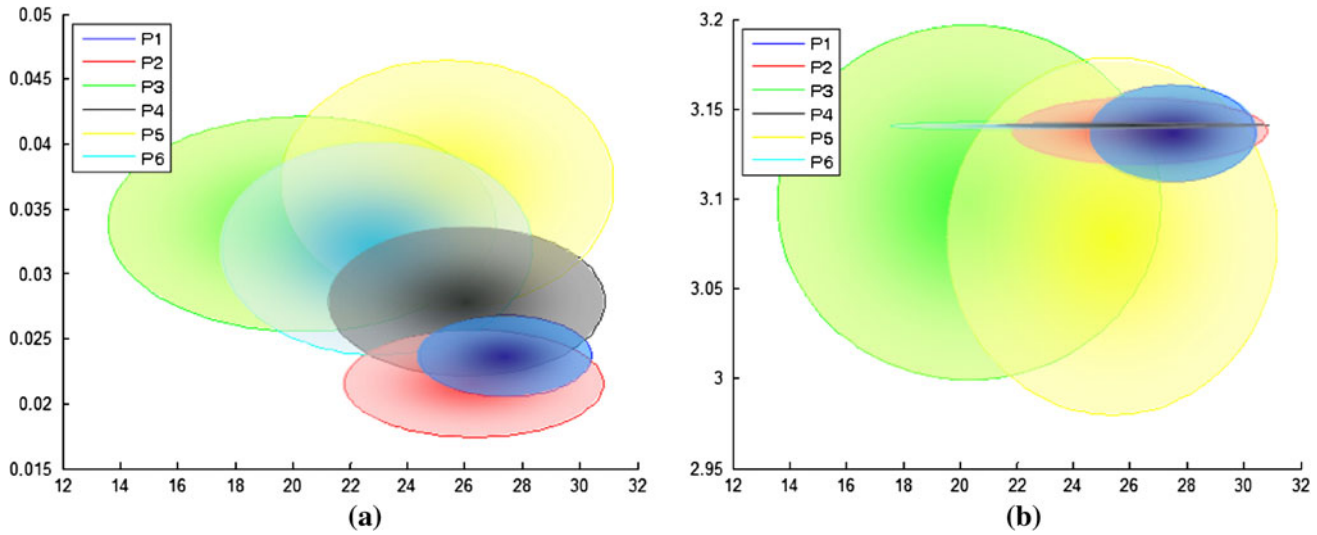

Fig. 11 Analysis of the first sine wave—mean and standard deviation of all six players. **a** Amplitude (a_1) versus angular frequency (b_1); **b** amplitude (a_1) versus phase (c_1)

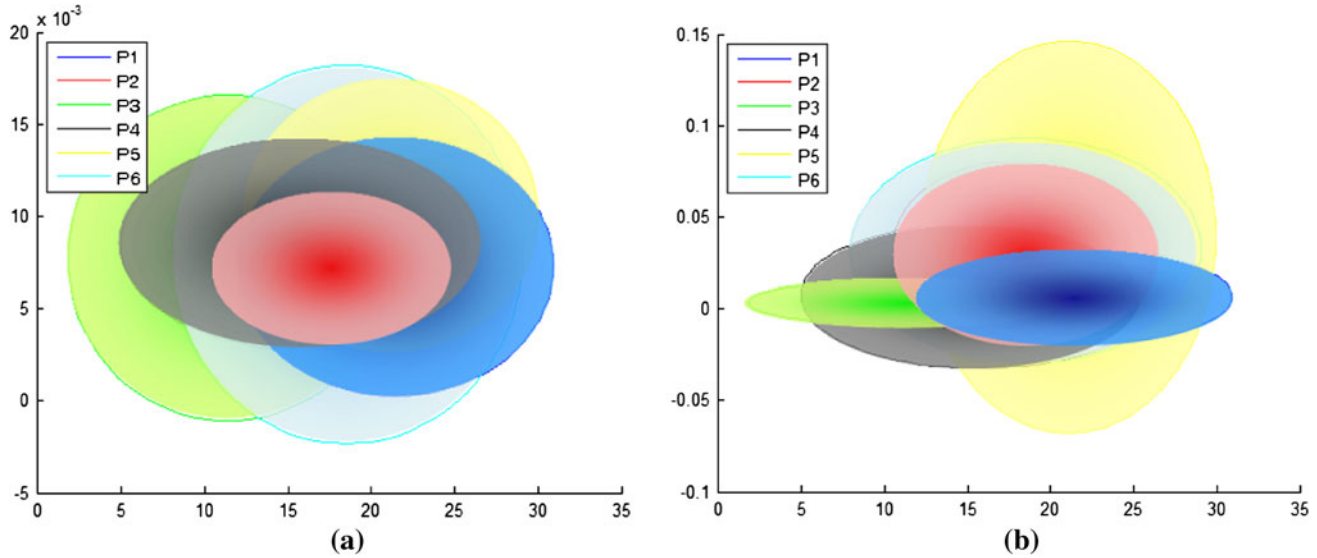

Fig. 12 Analysis of the second sine wave—mean and standard deviation of all six players. **a** Amplitude (a_2) versus angular frequency (b_2); **b** amplitude (a_2) versus phase (c_2)

It is noteworthy that the Rastrigin function presents a difficult problem due to the link between the size of the search space and the number of sub-optimal solutions. In fact, when attempting to solve Rastrigin function, most optimization algorithms easily fall into sub-optimal solutions. However, the DPSO is capable of maintaining a large diversity, thus yielding better results than PSO.

In these preliminary experiments, we were able to select the DPSO as a valid estimation algorithm and also to tune its parameters for the rest of the study presented in the next section.

6.2 Second stage: extracting the individual putting "signature"

In this stage of the work, intensive Matlab simulation was performed using the detection algorithm and the DPSO as an estimation technique, with the earlier defined parameters, to obtain the putter's motion function that describes 30 putt executions of six different expert subjects, in a total of 180 trials.

After calculating all the estimation parameters, some trials were removed to obtain a more representative sample

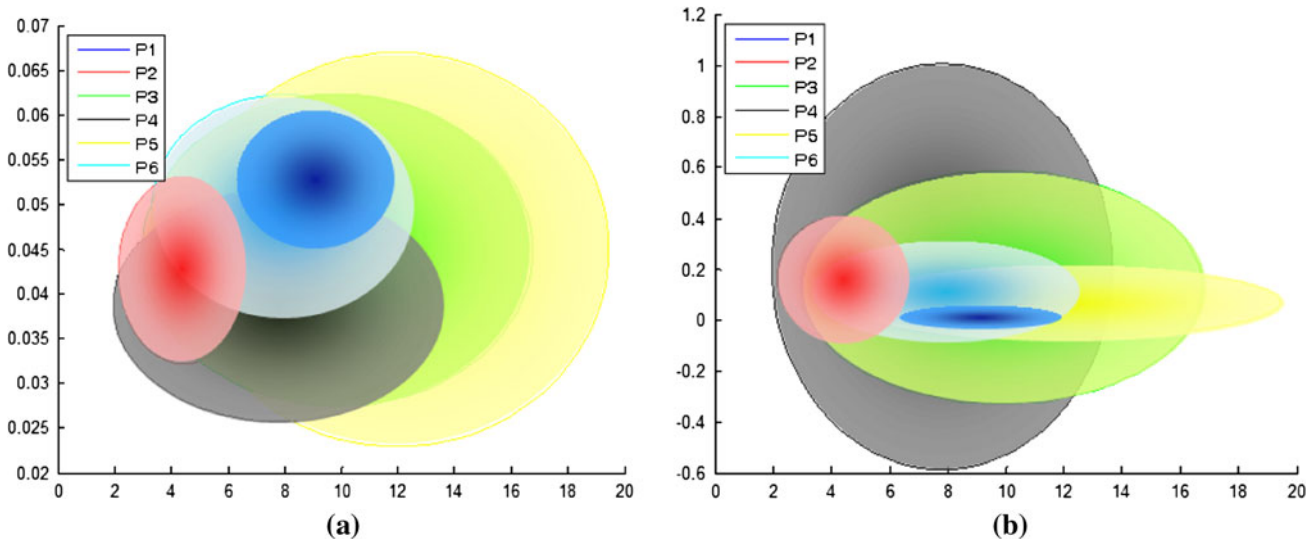

Fig. 13 Analysis of the third sine wave—mean and standard deviation of all six players. **a** Amplitude (a_3) versus angular frequency (b_3); **b** amplitude (a_3) versus phase (c_3)

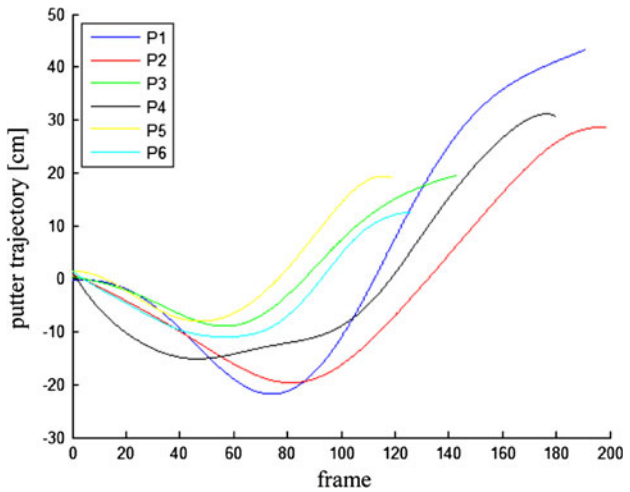

Fig. 14 Average putter trajectory of each subject

of each player through the identification of outliers in the following way:

- Each player's trials were studied by grouping the parameters of each of the three sine waves of the estimation function: amplitude versus angular frequency and amplitude versus phase, which resulted in six different graphics.
- The mean and standard deviation of each player's 30 trials were calculated, resulting in a unique ellipse for each pair of estimation parameters, as show in Figs. 11, 12 and 13.
- The trials that fell outside of each of the six representative ellipses were tracked. If a given trial fell outside at least three out of the six ellipses, then it is identified as an outlier and is therefore ignored.

After gathering all the inliers, which were typically around 15 trials for each player, the average putter's motion curves were computed. Such curves are presented in Fig. 14.

It can be verified in Figs. 11, 12 and 13 that each player's putting ability can be defined by six pairs of parameters related to the three sinusoids of the estimation model. Different players have different regions defined in the parameter's space, which in turn indicates a difference in the style of play by each subject. For instance, player 1 and 2 seem to have a more regular putt gesture, since the parameters present a smaller standard deviation (smaller regions) when compared with the other players. However, as expected, intersections between the ellipses in the parameter's space are common due to the player's expertise. Therefore, Fig. 14 represents the average putter trajectory of each subject, considering the inliers for all players and Table 2 the individual analysis of the corresponding actions during the movement. The information provided by them numerically identifies aspects like timing and stroke's width of the subjects in question, in order to differentiate them and extract a complete putting signature, which is discussed below.

Moving on to strokes' width, player 1 (represented in blue) exhibited, on average, the longest back stroke amplitude with 21.8 cm, whilst player 5 exhibited the shortest, with only 8 cm. It is interesting to notice that player 1, having larger back stroke amplitude, hit the ball with the highest average velocity. This is particularly obvious when compared with player 2, who has a lower amplitude (19.6 cm), but takes 51 video frames (around 0.24 s) to hit the ball, while player 1 only takes 39 frames (around 0.19 s) to hit the ball from a greater back stroke distance. In terms of forward

Table 2 Putting characteristics of every player

	Backswing (number of frames)	Neg. amplitude (cm)	Downswing (number of frames)	Ball impact (frame)	Follow-through (number of frames)	Pos. amplitude (cm)	Putt (number of frames)	Time (s)
P1	73	−21.8	39	112	79	43.3	191	0.91
P2	82	−19.6	51	133	66	28.6	199	0.95
P3	57	−8.9	28	85	58	19.5	143	0.68
P4	46	−15.1	71	117	63	31.1	180	0.86
P5	47	−8.0	30	77	42	19.4	119	0.57
P6	57	−11.0	36	93	33	12.5	126	0.60

strokes, the longest was also attained by player 1 with 43.3 cm and the shortest by player 6 with 12.5 cm.

Player 3 displayed high regularity in terms of timing, consistently performing near the median of the overall plays. In fact, the player presented a backswing and follow-through with similar timing (around 0.27 s) and an almost constant speed of 0.7 m/s in the downswing, as well as in the follow-through. These two characteristics contributed to the outstanding smoothness of the respective average curve, represented in green. As for the strokes' width, player 3 was between the players with the lowest back and forward stroke amplitude.

As it can be seen in the results, all players display unique putting characteristics. Similarities can be easily identified between them; however, the same can be stated about differences. These differences allow extracting a signature of each player, which enables us to affirm that a given putting stroke has a higher probability of belonging to a player instead of another.

7 Conclusion and future work

In the past decade, detailed biomechanical motion analysis has become an important part of athletic training and performance evaluation. More and more sensing devices such as cameras have been installed in environments where it is possible to automatically interpret and analyze intentional activities. Examples of such application domains are human living environments where computer systems are supposed to support people's everyday life, or factories where machines and human workers are supposed to perform production processes cooperatively.

Another domain that has been receiving increasing attention is the real-time automated analysis of sport games such as football, tennis or golf. Nowadays, in many live broadcasts, computer vision analysis, with special attention to the ball's kinematic, is used for example to present the ball's velocity or checking the ball's relative position. Also, more and more special importance to information computed off-line is given, like player's statistics. We consider the presented work as part of this framework, in the sense that real-time detection can be applied and similar off-line estimation information to the one presented in this article can be provided, in this case in the context of a live transmission of a golfing event.

The presented system for data retrieval, despite its complexity, is functional and allows retrieving a series of different information simultaneously. In this work, the putter's detection was carried out using a simple and computationally efficient computer vision algorithm, which presented good results. Also, a study of nonlinear estimation techniques was conducted and various approaches were tested to extract a sinusoidal function to model the putter's horizontal position in time.

The results confirmed the superior performance of the DPSO method, which was implemented based on the work of Tillett et al. [63], the first work to verify and apply the algorithm beyond the original authors, to our best knowledge.

With the implementation of the detection and estimation algorithms, this study benefits by using automatic tracking to analyze the putter movement in different stages (i.e., backswing, downswing and follow-through). To validate the work, toward using it in real situations, six expert golf players were tested. The experimental results clearly show when considering different trials of every player that each of them has a typical and distinct style of play. Properties of the average putter trajectory of each player, like the parameters of the estimation model, the timing of the play and its different stages, as well as the amplitude of the strokes, fully characterize the putting gesture of every player equivalently to a signature.

It is the authors' opinion that this study should not be confined to golfing plays. A similar study could eventually be done to analyze and extract signatures in the context of other sports like hand motion on swimming, boxing, tennis or baseball or even other body parts in everyday human actions.

In future work, we intend to relate each player's signature with the putt's degree of success using another camera to analyze the ball's trajectory, the error distances

in vertical length (VE), horizontal width (HW) and radial error (RE) to the hole. In this analysis, we will mainly focus on the pixel resolution of the camera instead of the frame rate, as we did in the camera that analyzed the putt gesture. Shooting at 30 fps (30 Hz) at a resolution of 1,280 × 720 pixels is considered adequate to capture the trajectory of the ball. This posterior study will eventually have the ability to automatically evaluate the quality of a given putt, considering all parameters.

Acknowledgments This work was supported by Ph.D. scholarships (SFRH/BD/73382/2010) and (SFRH/BD/64426/2009) by the Portuguese Foundation for Science and Technology (FCT), the Institute of Systems and Robotics (ISR) and RoboCorp at the Engineering Institute of Coimbra (ISEC) also under regular funding by FCT.

References

1. "Merriam-Webster Online". http://www.merriam-webster.com/ (last visited in August 2010)
2. Chiviacowsky S, Pinho TS, Alves D, Schild JFS (2008) Feedback autocontrolado: efeitos na aprendizagem de uma habilidade motora específica do golfe. Revista Brasileira de Educação Física e Esporte 22(4):265–271
3. Guadagnoli MA, Holcomb WR, Weber TJ (1999) The relationship between contextual interference effects and performer expertise on the learning of putting task. J Hum Mov Stud 37:19–36
4. Horner K, Fitzpatrick K, Smyth P (2008) The effect of increasing contextual interference on the practising of a motor skill". In: Cabri J, Alves F, Araújo D, Barreiros J, Diniz J, Veloso A (eds) Book of abstracts. 13th annual congress of the ECSS. Sport Science by the Sea, Estoril, p A-71
5. Maxwell JP, Masters RSW, Eves FF (2000) From novice to no know-how: a longitudinal study of implicit motor learning. J Sports Sci 18:111–120
6. Porter JM, Magill RA (2005) Practicing along the contextual interference continuum increases performance of a golf putting task. J Exerc Psychol 27:S-124
7. Hume PA, Keogh J, Reid D (2005) The role of biomechanics in maximising distance and accuracy of golf shots. Sports Med 35(5):429–449
8. Nesbit SM, Hartzell TA, Nalevanko JC, Starr RM (1996) A discussion of iron golf club head inertia tensors and their effects on the golfer. J Appl Biomech 12(4):449–469
9. Pelz D (1989) Putt like the pros. Harper Collins, New York
10. Pelz D (2000) Putting Bible: the complete guide to mastering the green. Publication Doubleday, New York
11. Porter JM (2008) Systematically increasing contextual interference is beneficial for learning novel motor skills. A dissertation submitted to the graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfilment of the requirements for the degree of doctor of philosophy. The Department of Kinesiology, pp 1–283
12. McCarty JD (2002) A descriptive analysis of golf putting: what variables affect accuracy? Master of Science thesis. Purdue University, USA
13. Mendes R, Martins R, Dias G (2008) Effects of a contextual interference continuum on golf putting task. Cabri J, Alves F, Araújo D, Barreiros J, Diniz J, Veloso A (eds) Book of abstracts. In: 13th annual congress of the ECSS. Sport Science by the Sea, Estoril, p. A-490
14. Perner P (2001) Motion tracking of animals for behavior analysis. In: Proceedings of the 4th international workshop on visual form. Lecture Notes in Computer Science, pp 779–786
15. Chen D, Yang J (2007) Robust object tracking via online dynamic spatial bias appearance models. IEEE Trans Pattern Anal Mach Intell 29:2157–2169
16. Batista J, Peixoto P, Fernandes C, Ribeiro M (2006) A dual-stage robust vehicle detection and tracking for real-time traffic monitoring. In: 9th international IEEE conference on intelligent transportation systems (ITSC), Toronto, pp 17–20
17. Javed O, Shah M (2003) KNIGHT: a multi-camera surveillance system. In: IEEE international conference on multimedia and expo 2003, Baltimore, pp 649–652
18. Gandhi T, Trivedi M (2007) Pedestrian protection systems: issues, survey and challenges. IEEE Trans Intell Transp Syst 8(3): 413–430
19. Gerónimo D, López AM, Sappa AD, Graf T (2010) Survey of pedestrian detection for advanced driver assistance systems. IEEE Trans Pattern Anal Mach Intell 32:1239–1258
20. Abdelkader M, Chellappa R, Zheng Q (2006) Integrated motion detection and tracking for visual surveillance. In: Proceedings of the 4th IEEE international conference on computer vision systems (ICVS 2006), pp 28–34
21. Cheng Y (1995) Mean shift, mode seeking and clustering. IEEE Trans Pattern Anal Mach Intell 17(8):790–799
22. Bradski G (1998) Computer vision face tracking as a component of a perceptual user interface. Workshop on Applications of Computer Vision, Princeton, pp 214–219
23. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 22(5):603–619
24. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):564–575
25. Pérez P, Vermaak J, Blake A (2004) Data fusion for tracking with particles. Proc IEEE 92(3):495–513
26. Fukunaga K, Hostetler L (1975) The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans Inf Theory 21(1):32–40
27. Chang C, Ansari R (2005) Kernel particle filter for visual tracking. IEEE Signal Process Lett 12(3):242–245
28. Maggio E, Cavallaro A (2005) Hybrid particle filter and mean shift tracker with adaptive transition model. In: Proceedings of ICASSP
29. Han B, Comaniciu D, Zhu Y, Davis LS (2004) Incremental density approximation and kernel-based bayesian filtering for object tracking. CVPR 1:638–644
30. Shan C, Tan T, Wei Y (2007) Real-time hand tracking using a mean shift embedded particle filter. J Pattern Recognit 40(7): 1958–1970
31. Cai Y, de Freitas N, Little JJ (2006) Robust visual tracking for multiple targets. In: Proceedings of European conference on computer vision, pp 107–118
32. Bai K, Liu W (2007) Improved object tracking with particle filter and mean shift. In: Proceedings of the IEEE international conference on automation and logistics, Jinan, vol 2, pp 221–224
33. Brasnett P, Mihaylova L, Canagarajah N, Bull D (2005) Particle filtering with multiple cues for object tracking in video sequences. In: Proceedings of SPIE's 17th annual symposium on electronic imaging, science and technology, V. 5685, pp 430–441
34. Brasnett P, Mihaylova L, Canagarajah N, Bull D (2007) Sequential Monte Carlo tracking by fusing multiple cues in video sequences. Image Vis Comput 25(8):1217–1227
35. Simon M, Behnke S, Rojas R (2000) Robust real time color tracking. In: 4th international workshop on RoboCup (robot world cup soccer games and conferences). Lecture Notes in Computer Science, pp 239–248

36. Pérez A, López F, Benlloch J, Christensen S (2009) Colour and shape analysis techniques for weed detection in cereal fields. Comput Electron Agric 69(1):73–79
37. Vezhnevets V, Sazonov V, Andreeva A (2003) A survey on pixel-based skin color detection techniques. Proc Graph 2003:85–92
38. Darrell T, Gordon G, Harville M, Woodfill J (2000) Integrated person tracking using stereo, color, and pattern detection. Int J Comput Vis 37(2):175–185
39. Yan F, Christmas W, Kittler J (2008) Layered data association using graph-theoretic formulation with application to tennis ball tracking in monocular sequences. IEEE Trans Pattern Anal Mach Intell 30(10)
40. Wolf JK, Viterbi AM, Dixon SG (1989) Finding the best set of k paths through a trellis with application to multitarget tracking. IEEE Trans Aerosp Electron Syst 25:287–296
41. Quach T, Farooq M (1994) Maximum likelihood track formation with the Viterbi algorithm. In: IEEE conference on decision and control, pp 271–276
42. Luber M, Arras KO, Plagemann C, Burgard W (2009) Classifying dynamic objects: an unsupervised learning approach. Auton Robots
43. Lucey S, Matthews I (2006) Face refinement through a gradient descent alignment approach. In: Proceedings of the HCSNet workshop on use of vision in human–computer interaction, Canberra, vol 56, pp 43–49
44. Momma M, Bennet K (2002) Pattern search methodology for support vector machines model selection. In: Proceedings of the SIAM international conference on data mining, Arlington
45. Zhou H, Seyfarth B (2005) A pattern search method for image registration. Lecture Notes in Computer Science, SpringerLink, vol 3514, pp 664–670
46. Emery L, Borland M, Shang H (2003) Use of a general-purpose optimization module in accelerator control. In: Proceedings of the 20th particle accelerator conference, Portland, p 2330
47. Miura K, Hashimoto K, Inooka H, Gangloff J, Matheli M (2006) Model-less visual servoing using modified simplex optimization. Artif Life Robot 10(2):131–135
48. Tang J, Zhu J, Sun Z (2005) A novel path panning approach based on AppART and particle swarm optimization. In: Proceedings of the 2nd international symposium on neural networks, LNCS, vol 3498, pp 253–258
49. Solteiro Pires EJ, de Moura Oliveira PB, Tenreiro Machado JA, Boaventura Cunha J (2006) Particle swarm optimization versus genetic algorithm in manipulator trajectory planning. In: 7th Portuguese conference on automatic control
50. Couceiro MS, Mendes R, Fonseca Ferreira NM, Tenreiro Machado JA (2009) Control optimization of a robotic bird. EWOMS'09, Lisbon
51. Alrashidi MR, El-Hawary ME (2009) A survey of particle swarm optimization applications in electric power systems. IEEE Trans Evol Comput 13(4):913–918
52. Couceiro MS, Luz JMA, Figueiredo CM, Ferreira NMF, Dias G (2010) Parameter estimation for a mathematical model of the golf putting. In: Proceedings of WACI'10. Workshop applications of computational intelligence 2010. ISEC.IPC, Coimbra, pp 1–8. ISSN 978-989-8331-10-6
53. Shi Y, Eberhart R (2001) Fuzzy adaptive particle swarm optimization. In: Proceedings of IEEE congress on evolutionary computation, vol 1, pp 101–106
54. Pires EJS, Machado JAT, Oliveira PBM, Cunha JB, Mendes L (2010) Particle swarm optimization with fractional-order velocity. J Nonlinear Dyn 61(295–301):2010
55. Blackwell T, Bentley P (2002) Don't push me! Collision-avoiding swarms. In: Proceedings of IEEE congress on evolutionary computation, vol 2, pp 1691–1696
56. Krink T, Vesterstrom J, Riget J (2002) Particle swarm optimization with spatial particle extension. In: Proceedings of IEEE congress on evolutionary computation, vol 2, pp 1474–1479
57. Miranda V, Fonseca N (2002) New evolutionary particle swarm algorithm (EPSO) applied to voltage/VAR control. In: Proceedings of the 14th power systems computational conference
58. Lovbjerg M, Krink T (2002) Extending particle swarms with self-organized criticality. In: Proceedings of IEEE congress on evolutionary computation, vol 2, pp 1588–1593
59. Chia-Feng J (2004) A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans Syst Man Cybern B Cybern 34(2):997–1006
60. Angeline P (1998) Using selection to improve particle swarm optimization. In: Proceedings of IEEE congress on evolutionary computation, pp 84–89
61. Zhang W, Xie X (2003) DEPSO: hybrid particle swarm with differential evolution operator. In: Proceedings of IEEE international conference on systems, man, and cybernetics, vol 4, pp 3816–3821
62. Kannan S, Slochanal S, Padhy N (2004) Application of particle swarm optimization technique and its variants to generation expansion problem. Electr Power Syst Res 70(3):203–210
63. Tillett T, Rao TM, Sahin F, Rao R (2005) Darwinian particle swarm optimization. In: Proceedings of the 2nd Indian international conference on artificial intelligence, Pune, pp 1474–1487
64. Couceiro MS, Figueiredo CM, Ferreira NMF, Machado JAT (2008) Simulation of a robotic bird. Fractional differentiation and its applications, Ankara
65. Knudson DV, Morrison CS (2002) Qualitative analysis of human movement. Human Kinetics Publishers
66. Paradisis G, Rees J (2000) Kinematic analysis of golf putting for expert and novice golfers. In: Proceedings of the 18th international symposium on biomechanics in sports, Hong Kong
67. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th international symposium on micro machine and human science, Nagoya
68. Yang X-S (2010) Test problems in optimization. In: Yang X-S (ed) Engineering optimization: an introduction with metaheuristic applications. Wiley, Hoboken