A Wireless Smart Shoe for Gait Analysis Using Heuristic Algorithms with Automated Thresholding

Nantawat Pinkam* and Itthisek Nilkhamhang

ABSTRACT

This paper proposes a novel decision system for segregation of five normal gait phases (stance, heel-off, swing 1, swing 2 and heel-strike) by using a real-time wireless smart shoe. The classification method employed four force sensitive resistors to measure the force underneath a foot, together with an inertial measurement unit that is attached at the back of the shoe to determine the magnitude of acceleration and the inclination angle of the foot with respect to the ground. Data acquisition was through the XBee wireless network protocol to allow serial processing by a computer. The state transition theorem and threshold-based classification were used to distinguish the gait phases according to received data, where the thresholds were optimized by heuristic algorithms, such as the genetic algorithm and particle swarm optimization. Ground truthing was determined by marker-tracking using image processing. Video recording with real-time data and embedded interfacing was used to verify the output of the proposed state transition algorithm under indoor testing on a stationary treadmill. Experimentation and verification were conducted on a subject with a normal gait cycle. The smart shoe was able to ascertain the gait phase with 96.07% accuracy after optimized threshold values had been determined.

Keywords: gait analysis, image processing, state transition theory, genetic algorithm, particle swarm optimization

INTRODUCTION

Ambulation is a fundamental component of human locomotion that enables a productive life. It is the recurring sequence of limb motion that concurrently moves the body forward and preserves the stability of the movement. In general human locomotion, one limb provides support while another limb moves to the next support site. The role of support is sequentially switched between the two limbs. When the limb is in the support phase, the foot is in contact with the ground to receive the body weight (Perry, 1992). The gait cycle of a normal person is efficient in terms of power consumption and movement speed, allowing the subject to walk with ease for an extended amount of time (Kong and Tomizuka, 2008).

Gait rehabilitation therapy is therefore essential for those people who have an abnormal gait. For example, Parkinson's disease causes the degradation of body movement and other motor abilities (Manap *et al.*, 2011). Elderly people suffer from abnormal gaits as a result of aging, leading to reduced mobility that can markedly impair their quality of life and increase the chance of serious

School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12121, Thailand.

* Corresponding author, email: pinkam.nantawat@gmail.com, itthisek@siit.tu.ac.th

Received date : 03/06/13 Accepted date : 11/09/13

injuries or, in some extreme cases, mortality caused by falling (Jahn *et al*., 2010). Transfemoral amputees also have trouble achieving a normal gait and although smart prosthetic devices have been developed to help achieve a natural walking cycle, the cost is prohibitively high (Torki *et al*., 2008). Currently, gait analysis utilizes various methods to assist people who have problems walking and attaining a normal gait (Marzani *et al*., 1997; Junho *et al*., 2008). In addition, the analysis can be used in sports to track the movement of athletes and increase their performance. Most importantly, gait analyzers can be used for the early detection of abnormalities in gait patterns, which could then be corrected by consulting with a physical therapist. Motion capture technology such as VICON (Vicon Inc., 2012) is an example of a gait rehabilitation device. It uses an infrared video camera with tracking markers to capture and reconstruct three dimensional body movements. Gait information can also be acquired through biomedical sensor technology, such as electromyography (EMG). EMG detects the electrical signals generated by skeleton muscle activity, in this case in the lower limbs, to verify the abnormality of the gait (Nissan *et al*., 2009). An example of an EMG-based gait rehabilitation device is the Locomat, which is used in locomotion therapy and was developed by Hocoma (Hocoma Inc., 2012).

This paper proposes a gait classification system using a wireless smart shoe, based on the force pattern underneath the foot measured by four force sensitive resistors (FSRs), and data from an inertial measurement unit (IMU). The IMU provides the magnitude of acceleration and the inclination angle of the foot. Gait phases are classified by a threshold-based state transition algorithm that is optimized using heuristic algorithms for increased accuracy, where ground truthing is determined by marker-tracking using image processing from recorded video. The data acquisition process is done wirelessly using a radio frequency module to allow for free movement while reducing the chances of wire entanglement or disconnection. The advantage of the proposed concept over the image processing approach is that there is no need for an additional machine to operate, there is increased mobility due to the small size of the device and the cost is cheaper when compared with a motion tracking system such as Vicon (Vicon Inc., 2013). In addition, the increased level of mobility allows the smart shoe to be used in other pieces of equipment, such as a prosthetic knee.

MATERIALS AND MEDTHODS

Definition of gait cycle

A normal gait is defined as the human walking pattern which results in minimum energy consumption and smooth motion (Kotaro and Richard, 2006). Gait cycle refers to a sequence of motion that repeats the same gait phases; for example, from the stance phase to the next stance phase of the same foot. It has been broadly divided into two phases; stance phase and swing phase (DeLisa, 1998). These phases can be further subdivided for more detailed gait analysis. Perry (1992) suggested that the gait cycle can be subdivided into the eight phases of initial contact, loading response, mid stance, terminal stance, pre swing, initial swing, mid swing and terminal swing. However, some consecutive phases are very similar. Therefore, this paper used a gait cycle consisting of five phases. A normal cycle starts from the stance phase, which is followed by heel-off, swing 1, swing 2 and heel-strike. After heel-strike, the cycle returns to the stance phase and begins again, as shown in Figure 1.

The following definitions of each gait phase are provided with reference to the shaded leg in Figure 1.

Stance phase

The foot sole of the shaded leg is mostly in contact with the ground and receives the body weight. This phase occurs after the heel strikes the ground or at the beginning of the gait cycle.

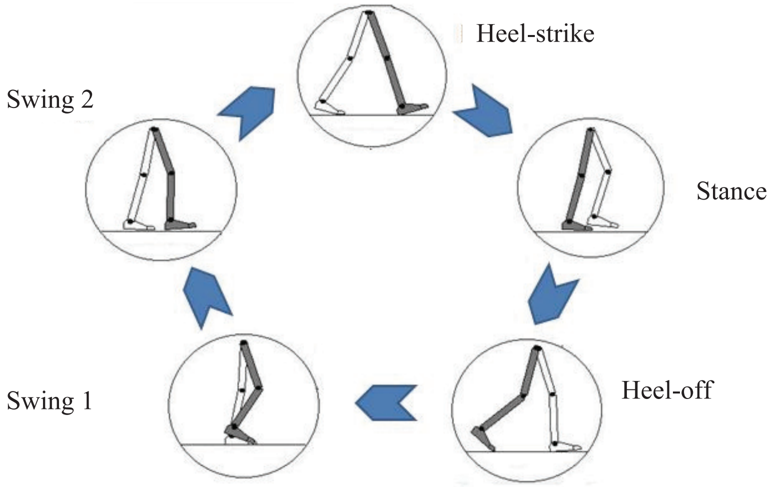

Figure 1 Normal gait cycle.

Heel-off phase

After the stance phase, the body begins to move forward. The heel of the shaded leg prepares to take off while the non-shaded leg starts touching the ground with its heel. The weight of the body is evenly distributed between the two legs.

Swing 1 phase

The full body weight passes to the non-shaded leg, whilst the shaded leg swings freely forward along the sagittal plane. The shaded leg is not in contact with the ground in this phase and the ankle remains behind the coronal plane. The sagittal and coronal planes are shown in Figure 2.

Swing 2 phase

The ankle of the shaded leg passes the coronal plane and continues to swing forward, causing the leg to become fully extended and ready for heel-strike.

Heel-strike phase

The end of the swing 2 phase results in the heel of the shaded leg touching the ground while the non-shaded leg prepares to move off the ground. This returns the body position to the stance phase and is the end of a normal gait cycle.

Construction of wireless smart shoe

While walking, force is exerted on the part of the foot that touches the ground. The smart shoe is constructed to measure this force underneath the foot by using force sensitive resistors (FSRs). The force is distributed across the entire sole of the foot, but there are certain locations where maximum force is applied to the bone structure. This corresponds to the location of FSR placement. However, during the swing phase, the force exerted on the foot sole provides unreliable data because the foot is not in contact with the ground but some forces remain as a

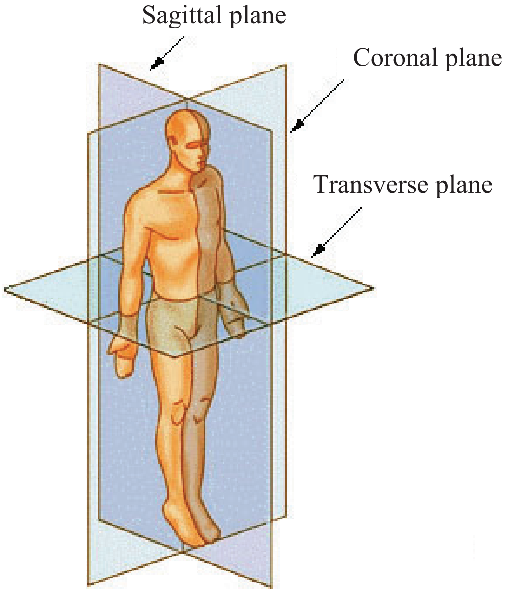

Figure 2 Illustration of body planes.

result of tension between the foot and the shoe. An inertial measurement unit (IMU), which is composed of a gyroscope, an accelerometer and a magnetometer, is therefore installed at the back of the shoe in a manner that minimizes disturbance while walking. The accelerometer provides the acceleration of the foot, and the foot inclination angles can also be calculated from IMU data. As the foot swings along the sagittal plane, only one particular angle is important. The IMU is positioned inside the box such that the roll angle corresponds to the sagittal plane.

An Arduino microcontroller (Arduino Uno R3; SmartProjects; Strambino, Italy) was used to acquire the raw data from the FSRs and IMU, which consisted of forces, the magnitude of acceleration and the foot inclination angle. The data were sent out through an XBee wireless communication module (XBee Pro Series 2; Digi International; Minnetonka, MN, USA) directly to a computer. Table 1 shows the complete list of components in a smart shoe. Figure 3 shows a schematic of the finalized sensor placements with an XBee radio frequency module installed. The prototype wireless smart shoe is shown in Figure 4. The normalized data measured by sensors during one gait cycle is represented by the graph in Figure 5. The foot inclination angle (Foot_Ang) is set to be negative if the slope of the foot is negative and it is positive when the slope is positive with respect to the ground.

Table 1 List of components.

Module	Component
Microcontroller	Arduino Uno R3
Force sensor	Force sensitive resistor–square
	Force sensitive resistor–0.5"
Inertial measurement unit	Pololu MinIMU-9 v2
Communication	XBee Pro Series 2

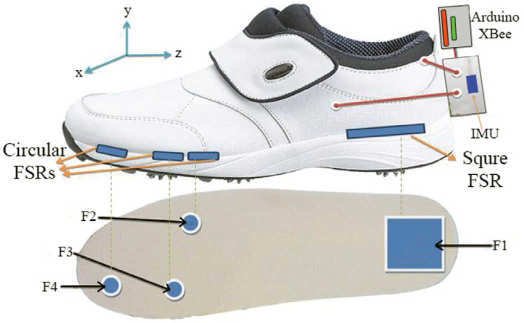

Figure 3 Sensor placement and XBee wireless module installation. (FSR = Force sensitive resistor; IMU = Inertial measurement unit; F1–F4 = Locations of FSRs.)

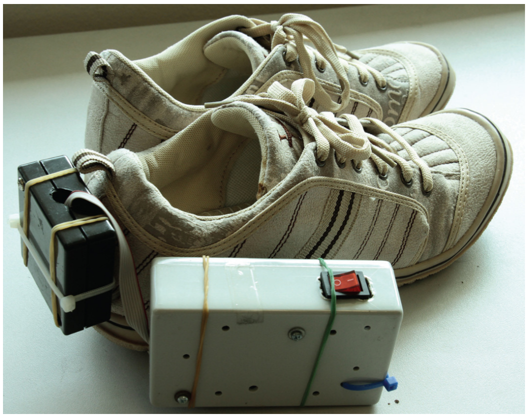

Figure 4 Prototype wireless smart shoe unit showing the inertial measurement unit strapped to the back of the shoe, with the XBee wireless communication module and Arduino contained in the lighter colored box.

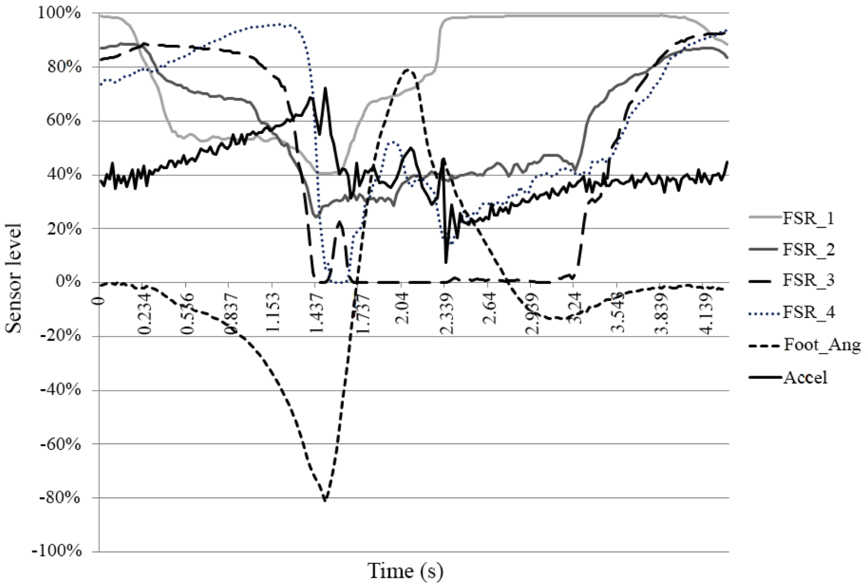

Figure 5 Normalized data of one gait cycle obtained wirelessly via the XBee wireless communication module. (FSR = Force sensitive resistor; Foot_Ang = Foot inclination angle; Accel = Acceleration.)

Gait information using marker-tracking

Before the wireless smart shoe can be used to classify gait phases, ground truthing must be determined from experimental data to verify the accuracy of the sensor outputs. The traditional method to determine gait phases is by manual observation and classification, which can lead to human error, while a more accurate alternative is the use of optical motion capture technology (OMC) for gait analysis (Guerra-filho, 2005). This is accomplished by requiring the test subject to wear a black body suit with markers placed on the body joints, as shown in Figure 6. The test subject is instructed to walk on a treadmill and is recorded by a high-frame rate camera. Sensor data is simultaneously collected via a wireless data acquisition system. This is followed by video processing which converts each frame of the video from an RGB channel to an HSI color model in order to distinguish the markers from other components. Then, the algorithm will determine the gait phase information by tracking the markers, as shown in Figure 6.

Gait phase classification using state transition theory

The extraction of gait phases for one ambulation cycle was completed by classification using the FSR, magnitude of acceleration and foot inclination angle data. A decision system was developed to classify the gait phases using state transition theory, based on the conditions of transition that were subdivided into sub-events and transition events. Thresholding was used to evaluate both sub-events and transition events.

The state transition algorithm was constructed by assigning each gait phase to a distinct state. Transition events were determined by signals received from the XBee radio frequency module, which included FSR, acceleration, and foot inclination angle data. FSRs provide the forces F1, F2, F3, and F4. The magnitude of acceleration was obtained from an accelerometer. The foot inclination angle was obtained from the IMU.

Tables 2 and 3 show sub-event conditions and transition events. They were constructed based on an observation of the changes in sensor data where a_{th1} and a_{th2} are the threshold values for the magnitude of acceleration. θ_{th1} and θ_{th2} are the threshold values for foot inclination angle. Sub-event conditions were derived by looking at the changes in the foot angle and acceleration data in each gait phase. G1 is the event when the foot is lying flat on the ground, such that its angle is within θ_{th1}. G2 is the event when the acceleration value is within the range of a_{th1} and a_{th2}. G3 is the event when the foot leaves the ground after the heel off phase, so its angle is lower than a_{th2}. G4 is the event after the swing 1 phase as the foot enters the swing 2 phase, so the angle is higher than a_{th2}.

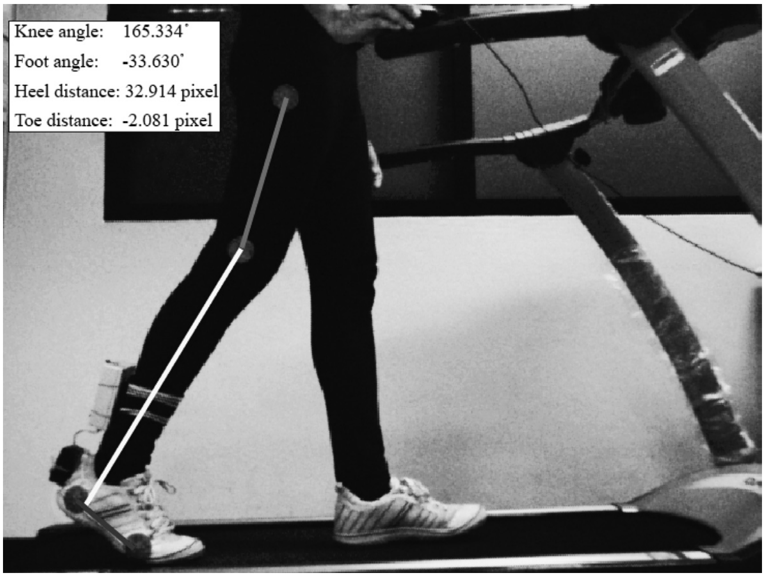

Figure 6 Gait phase information from marker-tracking.

Table 2 Sub-event conditions.

Sub-Event	Conditions
G1	$\lvert\theta\rvert < \theta_{th1}$
G2	$a > a_{th1}$ AND $a < a_{th2}$
G3	$\theta < \theta_{th2}$
G4	$\theta \geq \theta_{th2}$

a_{th1} and a_{th2} are the threshold values for the magnitude of acceleration; θ_{th1} and θ_{th2} are the threshold values for foot inclination angle.

Table 3 Transition events.

Transition Event	Conditions
E1	F1 = 1 AND G1 = 1 AND G2 = 1
E2	(F1 = 0 AND (F2 = 1 OR F3 = 1) AND G1 = 1) OR ((F4 = 1 OR F2 = 1 OR F3= 1) AND G1 = 0)
E3	F3 = 0 AND F4 = 0 AND G3 = 1
E4	F1 = 0 AND F3 = 0 AND G4 = 1
E5	F1 = 1 AND G4 = 1

F values indicate individual force sensitive resistors; G values represent events.

Transition event conditions were obtained by looking at the changes in FSR measurements and sub-event conditions together. Each FSR has its own threshold value to determine whether it is pressed (F = 1) or not pressed (F = 0). The threshold values of FSRs were applied separately for each sensor because the force exerted on the FSRs is not evenly distributed. E1 is a transitional condition to the stance phase. Force must be applied at F1 and the foot is flat on the ground with sub-event G1 and G2. E2 is a transitional condition to the heel off phase. The heel of the foot comes off the ground, so F1 = 0 and F2 or F3 = 1 but the foot's angle may not exceed G1. Another case is when F2 or F3 or F4 = 1 and the angle exceeds G1. E3 is the transitional condition to the swing 1 phase. F3 and F4 should not be pressed while the foot's angle undergoes sub-event G3. E4 is the transitional condition to the swing 2 phase. F1 and F3 should not be pressed while the foot's angle undergoes sub-event G4. E5 is the transitional condition to the heel strike phase. The foot strikes the ground heavily on the heel, such that F1 = 1 and its angle remains in sub-event G4. Figure 7 illustrates state transition according to the gait cycle and transition events.

Heuristic optimization algorithms

The traditional method to determine appropriate threshold values is to observe the change in sensor data from a large sample of experiments until acceptable results are found (Srivises *et al.*, 2012b). The accuracy of this method is questionable and it is difficult to prove that the thresholds are suitable for all test subjects. The current study introduced and compared two heuristic optimization methods: the genetic algorithm and particle swarm optimization.

Threshold set

The threshold set is defined as the threshold values that are input into both heuristic algorithms; in the genetic algorithm, it is called a chromosome, while in particle swarm optimization, it is called a particle. The threshold set consists of four FSR thresholds, two magnitude of acceleration thresholds (a_{th1},a_{th2}) and two foot inclination thresholds (θ_{th1},θ_{th1}). These thresholds are used by the fitness function along with the dataset and known output phase to determine the score.

Fitness function

The fitness function is used to evaluate how close the threshold set is to the optimum point. It is defined as the mean square error of

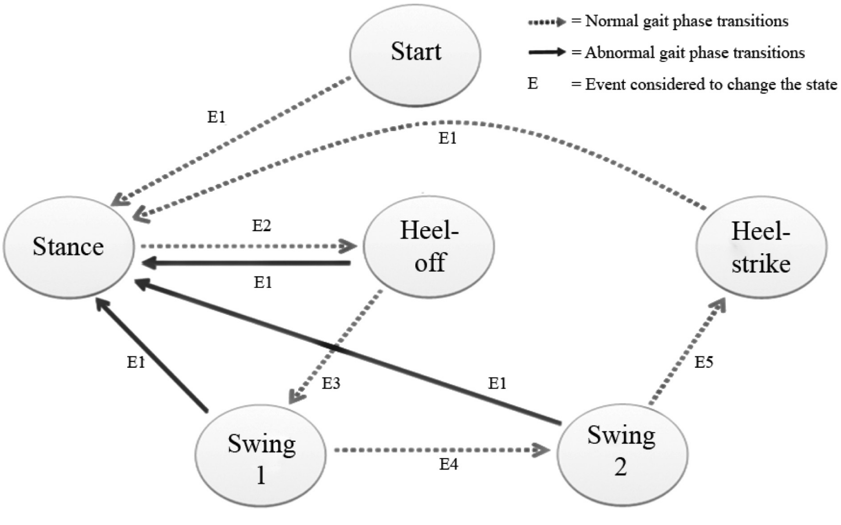
Start
Stance
Heel-off
Heel-strike
Swing 1
Swing 2
E1
E1
E2
E1
E1
E3
E1
E4
E5
= Normal gait phase transitions
= Abnormal gait phase transitions
E = Event considered to change the state

Figure 7 State transition diagram.

the misclassification distance between the visual classification output from a video sequence and the state transition output when the threshold values are applied to the dataset (Equation 1):

$$\text{Score} = E(\text{Normalized distance}(k,c)^2) \quad (1)$$

where k is a vector of the known output from the video sequence and c is a vector of the threshold set's output. The distance between the known output from the video and the threshold set's output is calculated based on the gait cycle in Figure 1. The distance function is defined as the number of steps required to transition from one gait phase to another. For example, if the actual state is stance but the threshold set's output is swing 1, the distance would be 2 because stance must pass through 2 states to reach swing 1. Moreover, the distance score is normalized by the number of sample points per state because each gait phase requires a different time duration. The notation E shows the mean of all squared distances. In this work, the same fitness function was utilized for both the genetic algorithm and particle swarm optimization.

Genetic algorithm

The genetic algorithm (GA) is a population-based model that uses selection and recombination to generate new sample points in a search space (Darrell, 1994). It is well known as a function optimizer tool (Patalia and Kulkarni, 2010). The GA mimics the natural selection scheme in which the fittest population survives to the next generation by producing offspring and undergoing mutation (Mathew, 2006).

The GA begins with a population with randomized chromosome assignments. The fitness function is applied to each chromosome to determine its score. A selected number of chromosomes with the highest fitness score will survive to the next generation. All chromosomes, including those that have survived, will mate with each other and produce offspring through the crossover operation. A small number of the population will experience mutation by randomly changing certain parts of their chromosomes, thus ensuring diversity. The steps repeat continuously until the stopping criteria are reached. Finally, the GA returns the chromosome that has the best fitness score through all the generations. Figure 8 shows the general flow chart of genetic algorithms.

Population size: The number of chromosomes effects both the computation time and population diversity. This research determined a suitable population size to be 200, according to Figure 9 where the average computation time and average best fitness score are plotted as the population size was experimentally varied from 10 to 1,000. Population sizes of 50 to 1,000 exhibited no substantial difference in their average best fitness scores. However, the standard deviation indicates that there is noticeable variation in achieved scores for population sizes between 50 and 100. Therefore, a population size of 200 was chosen as the optimal number because it offered a low computation time and consistently yielded the best fitness score.

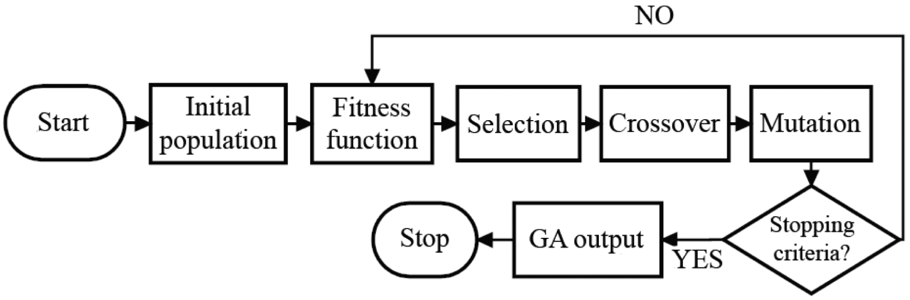

Figure 8 Genetic algorithm flow chart.

Selection: Each chromosome was evaluated by the fitness function, where lower scores signified less error and were considered to have a high fitness level. In contrast, higher scores indicated higher error and a lower fitness level. The scores were sorted into ascending order and fitness scaling was applied to convert the raw scores to a range of values that was suitable for selection. This research employed the rank scaling method as shown in Equation 2 (Mathwork Inc., 2012):

$$\text{Scale Value} = \frac{1}{\sqrt{n}} \qquad (2)$$

where n is the rank of the sorted score. The scale value was used as a probability weight in the selection process to create a mating table between two chromosomes. Stochastic universal sampling (SUS) according to Xu (1999) was employed as the selection operator. This method sampled the whole population with equally spaced pointers. Figure 10 shows how SUS operates. The outputs of SUS are chromosomes that have been chosen by the pointers.

Only 10% of the best fitness chromosomes were automatically passed to the next generation as elite members. The selected chromosomes from SUS will become the parents that are chosen to mate with each other at random.

Crossover: After selecting the mating pair, each parent must exchange some parts of their chromosomes in a process called crossover. This research employed a uniform crossover operation, where half of each parent's chromosome were randomly selected and swapped to create a next generation of chromosomes. At this point, the number of children was equal to the population size minus the elite members.

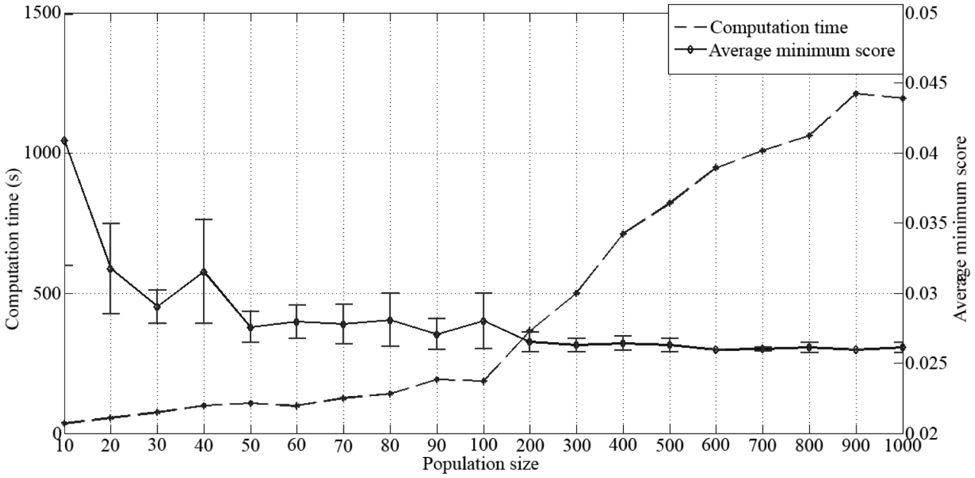

Figure 9 Average computation time and average minimum fitness score of the genetic algorithm. (Vertical bars indicate ± SD.)

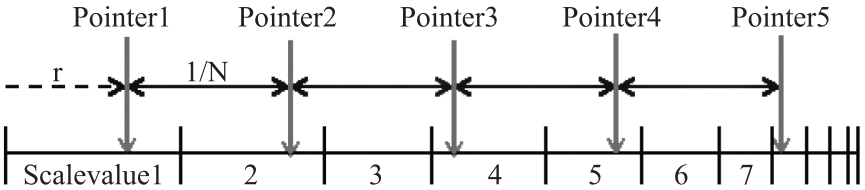

Figure 10 Stochastic universal sampling. (r is a random number, N is the population size and the step size of SUS is $\frac{1}{N}$.)

Mutation: 30% of the crossover children were randomly selected to experience mutation of their chromosomes. The function of mutation is similar to crossover, except that it will randomly change half of the chromosomes. The rate of change is inversely proportional to the generation number, thus ensuring that mutation will decrease as time passes and allow the population to converge on the most promising region.

Stopping criteria: The resulting behavior of the GA is that chromosomes with higher fitness scores are more likely to exchange information with each other to create new chromosomes that will have better fitness values than their ancestors. The process of the GA will stop when the stopping criteria are reached and the chromosome with the best fitness value is determined. The current study set the generation stall limit to 10, meaning that the GA will stop if no improvement in the best fitness score is achieved for 10 generations. Likewise, the process will stop once it reaches the maximum generation number of 100. These values were obtained through observation of the GA behavior when applied to experimental data.

Particle swarm optimization

Particle swarm optimization (PSO) is another tool to find the optimum threshold values for classifying the gait phases. It is a stochastic population-based global optimization algorithm inspired by the social behavior of fish schooling or bird flocking and was first introduced by Kennedy and Eberhart (Yang *et al.*, 2007). Each member of the swarm is called a particle and flies in the search space with a certain velocity. The velocity is updated by the influence of the particle's own experience and that of the entire swarm.

PSO begins with a number of particles randomly distributed in the search space. A fitness function is used to determine each particle's fitness score. Consequently, each particle will reassess its own best position and the best position located by the entire swarm. The velocity update rule was applied by using the previous velocity along with information of the particle's own best position and the global best position. The next position of the particle was calculated based on this new velocity. The algorithm then looped back to determine a fitness score until the stopping criteria were reached. Finally, PSO returned the position of the particle in the swarm that had the best overall fitness score. Figure 11 shows the flow chart of PSO.

Size of the swarm: The number of particles in the swarm must be chosen as a trade-off between the computation time and population diversity. This research determined a suitable swarm size according to Figure 12, which shows the average computation time and best fitness scores obtained by varying the population size from 10 to 1,000 and conducting five experiments. Population sizes 50 to 1,000 exhibited no substantial differences in the fitness score with acceptable standard deviation. Therefore, the optimum swarm size was chosen as 50 because it had a low computation time.

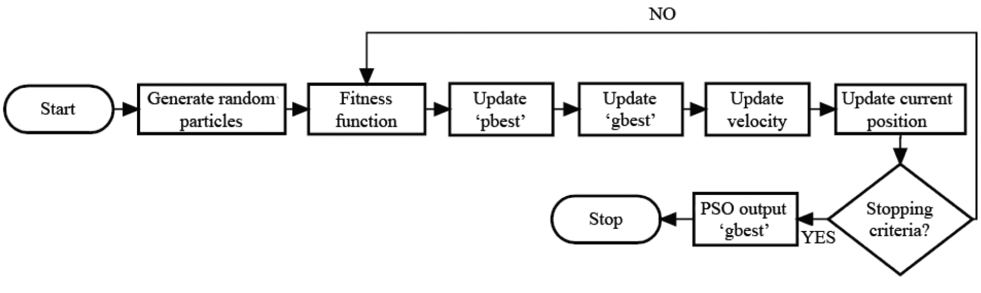

Figure 11 Particle swarm flow chart where pbest is the previous known best position of each particle and gbest represents the best known position in the swarm.

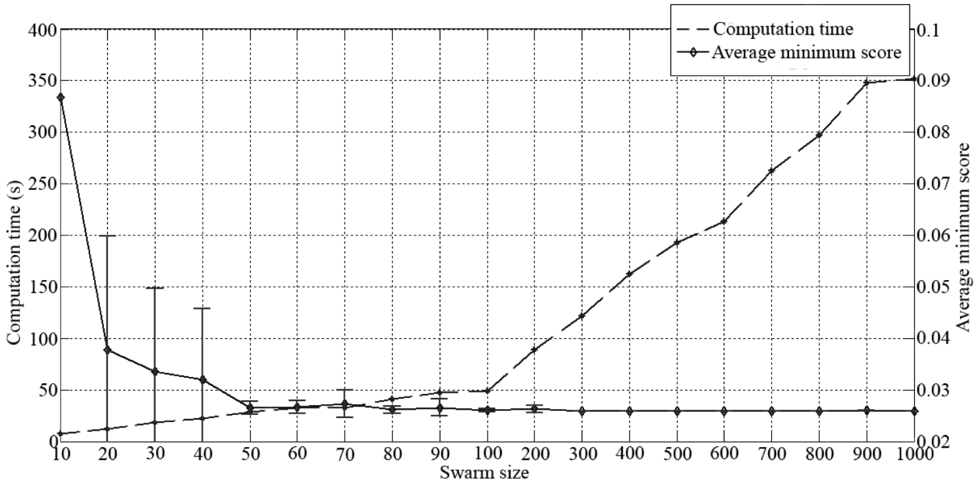

Figure 12 Average computation time and average minimum fitness score of particle swarm optimization. (Vertical bars indicate ± SD.)

Velocity update rule: After all the particles had been evaluated by the fitness function (where a low score signified less error and a high fitness, and a higher score indicated more error with low fitness), each particle's own best position and the global best position were updated, and each particle's velocity was calculated using Equation 3:

$$v_{id}^{t+1} = w \cdot v_{id}^{t} + c_1 \cdot \mathrm{rand}(\cdot)(\mathrm{pbest}_{id} - x_{id}^{t}) + c_2 \cdot \mathrm{rand}(\cdot)(\mathrm{gbest}_{d} - x_{id}^{t}) \quad (3)$$

where x is the position of the particle with corresponding velocity v, x_{id}^{t} signifies the i^{th} particle in dimension d at iteration *t* and the same notation applies to the velocity, pbest is the previous known best position of each particle, gbest represents the best known position in the swarm, w is the inertia weight factor applied to the velocity v_{id}^{t}, c_1 and c_2 are acceleration constants and rand(·) is a uniform random number in the range [0,1].

After the updated velocity was determined, it was added to the current position x^t to obtain the next position x^{t+1} according to Equation 4:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (4)$$

In this process, c_1 and c_2 acted as acceleration factors that pulled the particle toward the gbest and pbest positions, and were set from observation to 1 and 1.5, respectively. The inertia weight factor (w) was initially set as 0.9 and was gradually decreased to 0.4 by the final iteration (Eberheart and Shi, 2000), according to Equation 5 (Payakkawan *et al.*, 2009):

$$w = w_{max} - \frac{w_{max} - w_{min}}{t_{max}} \times t \quad (5)$$

where w_{max} = 0.9 is the maximum inertia weight and w_{max} = 0.4 is the minimum inertia weight, t denotes the current number of iterations and t_{max} is the maximum number of iterations.

Stopping criteria: The algorithm concluded when a certain stopping criterion was reached and it then output the best position of the swarm. PSO stops when there is no improvement in the gbest score for 10 iterations. In addition, if all particles converge to the gbest position with 10% difference in score value, the algorithm is stopped to save computation time. Otherwise, PSO stops upon reaching the specified maximum number of iterations.

RESULTS AND DISCUSSION

The sensor dataset was obtained from

a normal person walking on a treadmill with a smooth, even surface at a constant speed of 1 km.hr^{-1}. An Arduino Uno unit was employed to measure sensor signals, which were sent to a computer wirelessly via the XBee unit at a sampling rate of 50 Hz. A video camera with a frame rate of 60 frames per second was used to record the walk and provided ground truthing by allowing for visual classification of the gait phases based on image processing as discussed earlier. Several executions of the GA and PSO were performed to ensure the accuracy of the threshold values. Figures 13 and 14 show the average score and minimum score after the sensor dataset was inputted into the GA and PSO, respectively.

For a dataset with 4,792 sample points from 21 walking steps, the computation time was approximately 160 s for the GA and 51 s for PSO. Five-fold cross validation was used to determine the accuracy of the system. This method divided the dataset into five equivalent groups then performed the training and validation process five times, where the ith group was considered as the validation set in the ith round, while the rest of the data were used for training. The validation process indicated an accuracy of 95.93% for the GA and

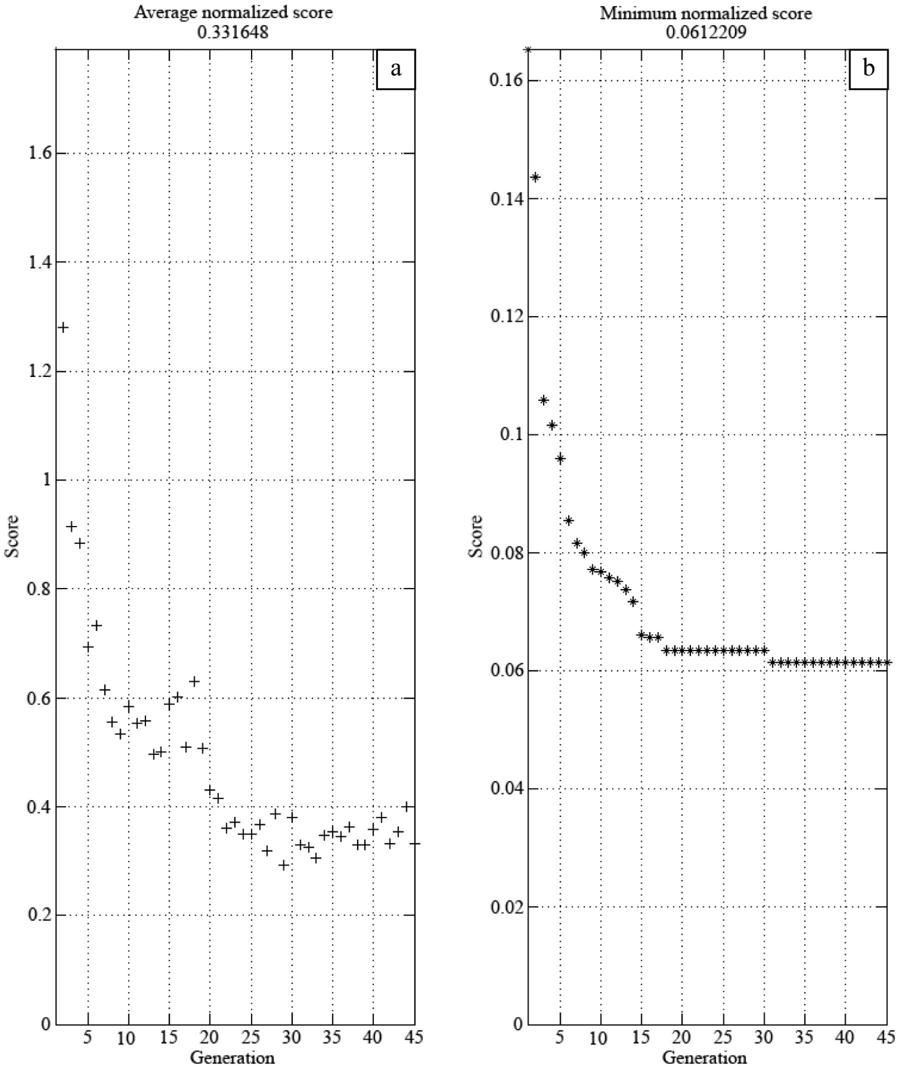

Figure 13 (a) Average fitness scores and (b) Best fitness scores for the genetic algorithm execution.

96.07% for PSO (Table 5). These threshold values obtained from both methods could then be used for real-time gait analysis with the wireless smart shoe.

The wireless smart shoe setup with automated thresholding using both methods was also compared with other systems based on state transition theory (Srivises *et al.*, 2012b) and fuzzy logic (Srivises, 2012a). The percentage accuracy was computed using Equation 6:

Accuracy = (Total data points – Miscalculated data points) / Total data points × 100% (6)

From Table 4, the proposed algorithm demonstrates substantial improvements in accuracy for all gait phases because in part, the optimized threshold values for transition events have been trained and validated using optimization tools for improved accuracy.

Table 5 shows a comparison between optimizing the threshold using the GA and PSO. The experimental results indicated that the GA required a larger population size to achieve the same accuracy level as PSO as well as being a more complicated algorithm with many

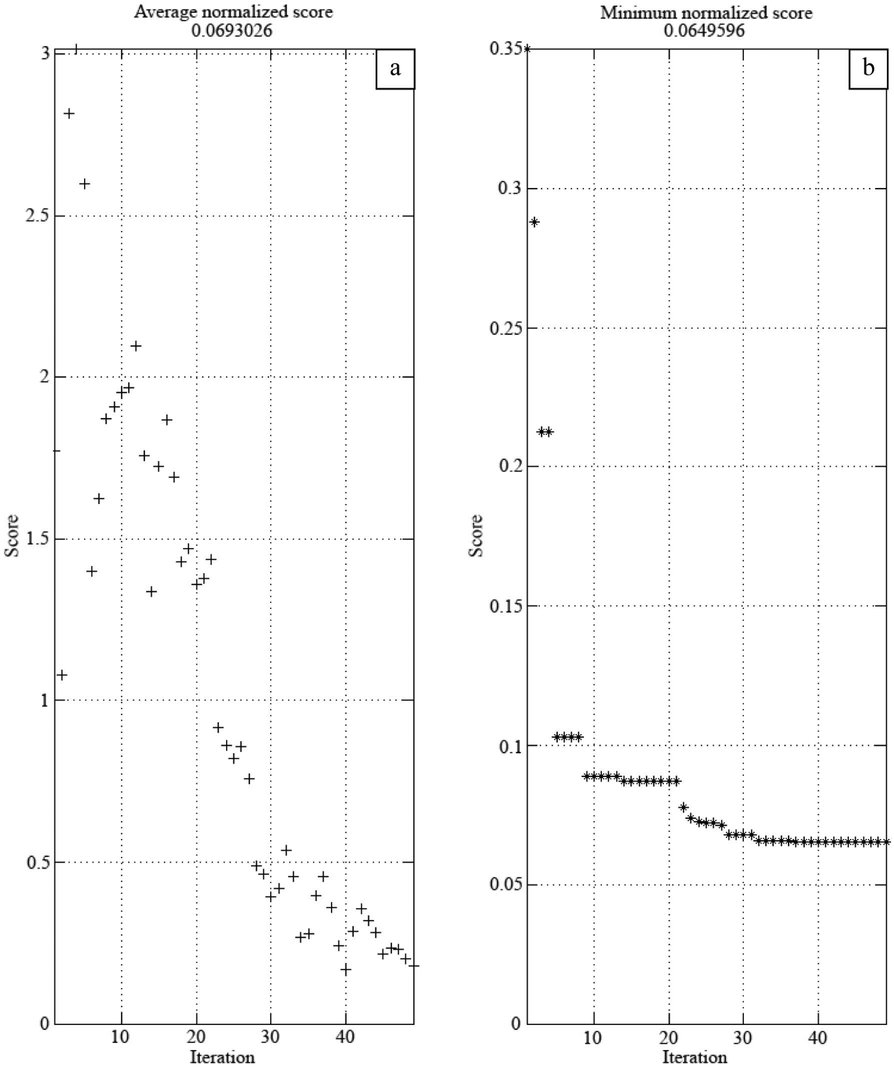

Figure 14 (a) Average fitness scores and (b) Best fitness scores for particle swarm optimization execution.

parameters to adjust, such as scaling, selection functions, crossovers and mutation rates. PSO was considerably easier to implement and required fewer design parameters, such as inertia weight and acceleration factors. It is well established that PSO is computationally faster than the GA, but it may become trapped by local optima (W. A. Lutfi *et al*., 2013). To prevent this, PSO should be executed several times to ensure the proper outcome. However, the achieved accuracy level of both the GA and PSO were similar.

CONCLUSION

A gait analyzer was presented that could distinguish between the different gait phases of stance, heel-off, swing 1, swing 2 and heel-strike. Force sensitive resistors were placed underneath the foot at strategic locations to measure contact with the ground. An inertial measurement unit was also installed at the back of the shoe to evaluate the foot inclination angle and the magnitude of acceleration. These data were transmitted to a computer via a wireless connection using an XBee radio frequency module. State transition theory was employed as the decision system for the gait phase classification, with threshold values determined by the genetic algorithm and particle swarm optimization. The genetic algorithm was preferable due to the fact that particle swarm optimization may be trapped by local optima. Ground truthing of the system was obtained by marker-tracking using image processing. The result was experimentally compared with conventional methods and showed a substantial improvement in accuracy.

ACKNOWLEDGEMENTS

This research was funded by the National Science and Technology Development Agency (NSTDA) of Thailand through the Young Scientist and Technologist Program (YSTP: SP55-NT06), the National Research University Project of

Table 4 Accuracy comparison.

Gait phase	Accuracy of state transition approach with gyroscope (%)	Accuracy of fuzzy logic approach with gyroscope (%)	Accuracy of Img-GA Opt. state transition with IMU (%)	Accuracy of Img-PSO Opt. state transition with IMU (%)
Stance	86.17	91.94	95.47	95.51
Heel-off	81.12	79.52	96.84	97.07
Swing 1	85.92	79.38	95.53	94.75
Swing 2	90.25	80.69	95.24	95.00
Heel-strike	68.42	93.33	95.15	95.89
Overall accuracy	82.10	85.10	95.93	96.07

Img-GA Opt. = Image processing and GA optimization; Img-PSO Opt. = Image processing and PSO optimization.

Table 5 Optimization algorithm comparison.

Criterion	Genetic algorithm	Particle swarm optimization
Population size	200	50
Implementation	Complicated with many parameters	Simple with few parameters
Computation time	160 s	51 s
Stopping iteration	38	43
Accuracy	95.93%	96.07%

Thailand Office of Higher Education Commission, and the Center of Excellence in Biomedical Engineering of Thammasat University.

LITERATURE CITED

Darrell, W. 1994. **A Genetic Algorithm Tutorial.** Colorado State University. Fort Collins, CO, USA. 37 pp.

DeLisa, J.A. 1998. **Gait Analysis in the Science of Rehabilitation.** Veterans Health Administration, Rehabilitation Research and Development Service, Scientific and Technical Publications Section, Department of Veterans Affairs. Pennsylvania, PA, USA. 112 pp.

Guerra-filho, G.B. 2005. Optical motion capture: Theory and implementation. **Journal of Theoretical and Applied Informatics (RITA)** 12: 61–89.

Hocoma Inc. (2012). **Lokomat.** [Available from http://www.hocoma.com/products/lokomat/]. [Sourced: 31 October, 2012].

Jahn, K. and A. Zwergal and R. Schniepp. 2010. Gait disturbances in old age—classification, diagnosis, and treatment from a neurological perspective. **Dtsch. Arztebl. International** 107: 306–316.

Junho, P., J. Choi and J. Cho. 2008. Development of gait phase analyzing devices for the gait training equipment for hemiplegia patients, pp. 632–636. *In* ICST, (eds.). **Proceedings of International Conference on Sensing Technology 2008.** 30 November–3 December 2008. ICST. Tainan, Taiwan.

Kong, K. and M. Tomizuka. 2008. Smooth and continuous human gait phase detection based on foot pressure patterns, pp. 3678–3683. *In* ICRA, (eds.). **Proceedings of IEEE International Conference on Robotics and Automation 2008 (ICRA 2008).** 19–23 May 2008. ICRA. California, CA, USA.

Kotaro, S. and R.N. Richard. 2006. Muscle mechanical work and elastic energy utilization during walking and running near the preferred gait transition speed. **Gait & Posture** 23: 383–390.

Manap, H.H., N.M. Tahir and A.I.M. Yassin. 2011. Statistical analysis of Parkinson disease gait classification using Artificial Neural Network, pp. 60–65. *In* ISSPIT, (eds.). **Proceedings of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) 2011.** 14–17 December 2011. ISSPIT. Bilbao, Spain.

MathWorks Inc. (2012). **Global Optimization Toolbox User's Guide.** [Available from http://www.mathworks.com/help/releases/R13sp2/pdf_doc/optim/optim_tb.pdf]. [Sourced: 31 October 2012].

Mathew, T.V. 2006. **Genetic Algorithm.** Institute of Technology Bombay. Mumbai, India. 15 pp.

Marzani, F., Y. Maliet and L. Dusserre. 1997. A computer model based on superquadrics for the analysis of movement disabilities, pp. 1817–1820. *In* IEEE, (eds.). **Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 1997.** 30 October–2 November 1997. IEEE. Chicago, IL, USA.

Nissan, K., K. Neelesh, P. Dinesh, D. Aseem and K. Amod. 2009. EMG signal analysis for identifying walking patterns of normal healthy individuals. **Indian Journal of Biomechanics** Special Issue (NCBM 7–8 March 2009): 118–112.

Patalia, T. P. and G. R. Kulkarni. 2010. Behavioral analysis of genetic algorithm for function optimization, pp. 1–5. *In* IEEE, (eds.). **Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) 2010.** 28–29 December 2010. ICCIC. Coimbatore, India.

Perry, J. 1992. **Gait Analysis: Normal and Pathological Function.** Slack Incorporated. New Jersey, NJ, USA.

Srivises, W., I. Nilkhamhang and K. Tungpimolrut. 2012a. Design of a smart shoe for reliable gait analysis using fuzzy logic, pp. 834–838. *In* SICE, (eds.). **Proceedings of SICE Annual Conference (SICE) 2012.** 20–23 August 2012. SICE. Akita, Japan.

________. 2012b. Design of a smart shoe for reliable gait analysis using state transition theory, pp. 1–4. *In* ECTI-CON, (eds.). **Proceedings of Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) 2012.** 16–18 May 2012. ECTI. Hua Hin, Thailand.

Torki, A.A., M.F. Taher and A.S. Ahmed. 2008. Design and implementation of a swing phase control system for a prosthetic knee, pp. 1–4. *In* CIBEC, (eds.). **Proceedings of IEEE Biomedical Engineering Conference 2008.** 18–20 December 2008. CIBEC. Cairo, Egypt.

Vicon Inc. (2012). **VICON.**[Avialable from http://www.vicon.com]. [Sourced: 31 October 2012].

________. (2013). **Bonita.** [Available from http://www.vicon.com/System/Bonita]. [Sourced: 31 August 2013].

W. M. Lutfi, W. M. H., C.S. Lim, A. F. Z. Abidin, M.H. Azizan and S.S. Teoh. Solving maximal covering location with particle swarm optimization. **International Journal of Engineering and Technology (IJET)** 5: 3301–3306.

Xu, H. 1999. **Comparison of Genetic Operators on a General Genetic Algorithm Package.** Faculty of the Graduate College of the Oklahoma State University. Stillwater, OK, USA. 78 pp.

Yang, X., J. Yuan, J. Yuan and H. Mao. 2007. A modified particle swarm optimizer with dynamic adaptation. **Applied Mathematics and Computation** 189: 1205–1213.