Metaheuristic Optimization Techniques for Articulated Human Tracking

by

Sohail Akhtar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Four adaptive metaheuristic optimization algorithms are proposed and demonstrated: Adaptive Parameter Particle Swarm Optimization (AP-PSO), Modified Artificial Bat(MAB), Differential Mutated Artificial Immune System (DM-AIS) and hybrid Particle Swarm Accelerated Artificial Immune System (PSO-AIS). The algorithms adapt their search parameters on the basis of the fitness of obtained solutions such that a good fitness value favors local search, while a poor fitness value favors global search. This efficient feedback of the solution quality, imparts excellent global and local search characteristic to the proposed algorithms.

The algorithms are tested on the challenging Articulated Human Tracking (AHT) problem whose objective is to infer human pose, expressed in terms of joint angles, from a continuous video stream. The Particle Filter (PF) algorithms, widely applied in generative model based AHT, suffer from the 'curse of dimensionality' and 'degeneracy' challenges. The four proposed algorithms show stable performance throughout the course of numerical experiments. DM-AIS performs best among the proposed algorithms followed in order by PSO-AIS, AP-PSO, and MBA in terms of Most Appropriate Pose (MAP) tracking error.

The MAP tracking error of the proposed algorithms is compared with four heuristic approaches: generic PF, Annealed Particle Filter (APF), Partitioned Sampled Annealed Particle Filter (PSAPF) and Hierarchical Particle Swarm Optimization (HPSO). They are found to outperform generic PF with a confidence level of 95%, PSAPF and HPSO with a confidence level of 85%. While DM-AIS and PSO-AIS outperform APF with a confidence level of 80%. Further, it is noted that the proposed algorithms outperform PSAPF and HPSO using a significantly lower number of function evaluations, 2500 versus 7200.

The proposed algorithms demonstrate reduced particle requirements, hence improving computational efficiency and helping to alleviate the 'curse of dimensionality'. The adaptive nature of the algorithms is found to guide the whole swarm towards the optimal solution by sharing information and exploring a wider solution space which resolves the 'degeneracy' challenge. Furthermore, the decentralized structure of the algorithms renders them insensitive to accumulation of error and allows them to recover from catastrophic failures due to loss of image data, sudden change in motion pattern or discrete instances of algorithmic failure. The performance enhancements demonstrated by the proposed algorithms, attributed to their balanced local and global search capabilities, makes real-time AHT applications feasible.

Finally, the utility of the proposed algorithms in low-dimensional system identification problems as well as high-dimensional AHT problems demonstrates their applicability in various problem domains.

Acknowledgements

In the name of Allah, Most Gracious, Most Merciful.

All Praises to Allah Almighty for giving me the ability and serenity to conduct this research, strength to persevere through hard times and for showering His countless blessings on me.

Many people have been overwhelmingly supportive and encouraging of this work that surely my gratitude cannot be adequately expressed in just words here. Without the guidance, help, and support of those around me, completing this research would not have been possible.

First and foremost, I want to pay my gratitude to my advisors, Prof. Eihab Abdel-Rahman and Prof. Abdul-Rahim Ahamd. Their guidance and input have been tremendously valuable and this research would not have been possible without them. I am also thankful for the knowledge, depth and breadth of perspectives and diversity of ideas they brought to enrich my research enterprise. Their manner of mentoring and supervision are also paramount. They both provide guidance and leadership but in a way that the student is free to try new techniques and experiments while ultimately managing his own research. Their leadership has greatly impacted my scientific thinking and developed my writing and analytical skills. They gave me the sense of professionalism: maintain the quality of work without indulging into unnecessary complications, which kept me focused in achieving my thesis objectives when it was easy to get lost during the exploration of interesting research ideas which popped every now and then. Some of these ideas are related to the practical applications of research to other fields such as controls and optimization, which we have started to discuss and explore now.

I must also mention the time they spared for me; the endless office meetings with Prof. Eihab and hours long phone and Skype conversations with Porf. Abdul-Rahim in the middle of the night that helped me understand intricate issues of my research. I foresee them as my research collaborators for many years to come and benefit from having a lasting relationship with them. Both of them are as meritorious a human being as a researcher. They were there for multiple hardships in my personal life and I cannot imagine going through them without their guidance and help, thanks for the patience shown during those hard times. I am also thankful for the generous financial support during this research. I consider it an honor for myself to be their student.

I would also like to thank my extraordinary dissertation committee, Prof. John Zelek, Prof. Keith Hipel and Prof. Otman Basir. I am so grateful that you each took time out

to be part of my thesis committee and guide me on this journey. Special thanks to Prof. John Zelek for enhancing my interest to the field of Monte Carlo simulations and helping me understand the intricacies of image processing and video tracking. I had a great time attending his course on Particle Filters.

I am thankful to Prof. Keith Hipel for providing me invaluable insight on broader systems design perspective in decision making. I will be staying in touch with him to explore societal aspects and implications of my research and ways of minimizing such issues in case of potential commercialization of the technologies we are developing.

Thanks to Prof. Otman Basir for his excellent lectures that gave insight to the area of soft computing and intelligent systems. I thoroughly enjoyed his style of teaching. His insights on the commercial value of applications of this research broadened my perspective beyond my initial motive of pursuing this research direction. I am also thankful to Prof. Richard Wildes for serving as the external member for my thesis defence committee.

Thanks are also due to the administrative staff of the Systems Design Engineering Department for their friendly and reliable support throughout the years. Special thanks are due to Late Tariq Naqvi for tremendously helping me at one of the hardest junctions of my academic life. Thanks to my friends and colleagues at the campus for providing a friendly and creative atmosphere. I met so many talented and wonderful individuals that are always ready to help and I will always cherish my time here.

Finally, I would like to express my deepest gratitude to my parents and my siblings for their prayers and encouragement, even from distance. I extend my sincerest gratitude to my wife, Afshan, for her love, patience, sacrifices, support and encouragement which made it possible to complete this research. I must also mention my son Arham and daughters Manaal and Anaum, who bring so much joy to my life and remain a source of love, motivation and inspiration throughout my doctoral roller-coster ride.

The research is partially funded by NSERC Postgraduate Doctoral Scholarship, Ontario Scholarship in Science and Technology (OGSST) and Research Studentship provided by my supervisor, which is gratefully acknowledged. I would also like to acknowledge the research grant from the Rowe School of Business, Dalhousie University, as well as NSTIP Grant#13-INF1094-10 of Government of Saudi Arabia.

Dedication

To Ammi, Abbu and my family
for their endless love, support and encouragement

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The focus of this dissertation is to decipher the challenging problem of full-body human tracking through metaheuristic based optimization. This chapter describes the motivations of this research as well as the opportunities and challenges involved in this complex application domain. Section 1.1 specifies the motivations of this research work. Section 1.2 explains the general approach for object tracking in image sequences. Section 1.3 investigates the particulars of Articulated Human Tracking (AHT). Section 1.4 describes the research challenges and sets out the problem statement, research goals and research contributions. Finally, Section 1.5 outlines the structure of this thesis.

1.1 Motivation

Human motion capture and analysis techniques cover a broad field of applications ranging from digital animation to biomechanical analysis of sports activities to clinical application in patient rehabilitation. The 3D video animation and tracking has a huge market that is expanding rapidly. It is expected to grow yearly at a rate of ∼13% from 2011 to 2016 and will worth $242.93 billions (according to marketsandmarkets.com) by the end of 2016. The market includes e-education, web design, animation for entertainment (movies, visual effects, 2D/3D animations), gaming (PC games, mobile phone games, console games) etc. Further, it is extensively used in intelligent video surveillance and airport security that is expected to capture a market share of $21 billions (according to marketsandmarkets.com) by 2020. In daily life, the application of gesture recognition is seen in modern day smart TVs, mobile phones and tablets. In short, human motion capture and analysis has enormous applications that are growing by the day.

Existing human motion capture and analysis systems use reflective markers that are attached to the subject [101, 74]. Although this method is popular and has high precision and robustness, but it has several limitations. Attached markers influence the subject's movement, consciously or subconsciously. A controlled environment is required to acquire accurate high quality data. It is time consuming to attach markers to the subject. The markers placed on the skin, also move relative to the bones/joints resulting in skin artifacts.

The cumbersome, time-consuming and invasive nature of marker-based systems has limited their use. An ideal measurement system should neither be invasive nor cumbersome. On top of this, measurements should also be taken in a natural environment, such as work place, home, sports field, with a sufficiently large field of view so that the subject's activity is not consciously or subconsciously compromised.

Recently, there has been a wide availability and development of data acquisition devices, such as video cameras, and a vast improvement in the computational capacity of computers and a huge reduction in hardware costs. Thus, the development of markerless measurement systems based on video data is an important step ahead in the direction of a non-invasive measurement and analysis system.

Markerless motion capture systems originate from the fields of computer vision and machine learning, where it has many applications such as human computer interface, surveillance, virtual reality, computer animation, gesture recognition and content based video indexing and retrieval. The analysis of markerless high-dimensional articulated human motion is a challenging and active research area. It has enormous potential and applications in various fields such as information technology, medical technology, sports, computers and security.

1.2 General Approach for Object Tracking

Object tracking can be defined as the method of following an object through successive image frames to determine its motion with respect to other objects [91]. Every tracking system has to answer the following key questions:

- Which object representation is suitable for tracking?
- Which image features should be used?
- How should motion, appearance, and shape of the object be modeled?

The block diagram in Figure 1.1 depicts a simplified bottom-up approach for the steps involved in building an object tracking system. The first step is the detection of object of interest in the current video frame. In fact, every tracking algorithm is required to detect the object of interest either in the first frame or in all the frames. Poor detection of a moving object has a detrimental effect on the overall performance of the tracking system. Background subtraction is usually used to extract the region of interest. The second step is a suitable representation of the detected object. For instance, an articulated human body is represented by connected shapes, such as ellipses, rectangles, cones, which represent human limbs. The next step is the selection of discriminative features that can be used for tracking over the course of video sequence. Edges, silhouette, color histograms and texture are the most commonly employed image features. The last step is the use of a suitable tracking/prediction algorithm. For articulated object tracking, Particle Filter (PF) is used quite often.

This research focusses on the prediction and refinement of human pose in video sequences and thus falls in the last block of the discussed object tracking computational framework.

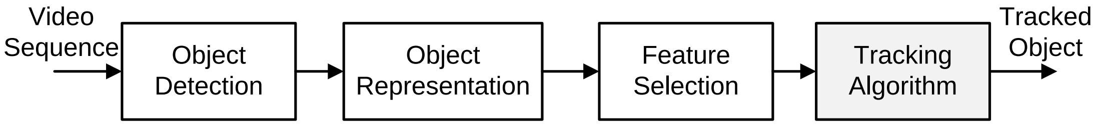


Figure 1.1: Bottom-up approach for video tracking

1.3 Challenges in AHT

Humans move in a very diversified manner. Both their appearance and motion varies with time. In addition, two different subjects in the same pose seldom appear similar. The situation complicates further as the depth information is lost due to the projection of 3D world onto a 2D image plane. Thus, extraction of 3D human motion from video sequences is a challenging problem. The following sections discuss some of the major challenges encounter in designing an AHT system.

1.3.1 High Dimensionality and Motion Constraints

Kinematic representation of humans consists of a skeletal structure consisting of cones, cylinders or globally deformable surfaces. A simple representation of a human, where limbs are represented by rigid elements connected by joints, requires 25 dimensions, such as states representing the joint angles. A more detailed representation may need up to 100 dimensions. Using such a high dimensional model for tracking requires estimating 25 to 100 parameters at each time step which is computationally very expensive [87]. Some kind of approximation or problem-dependent heuristics (temporal coherency, dynamical models, motion symmetries) are used in such situations [119]. Some data driven dimensionality reduction approaches are also employed to capture and synthesize the variability of human shape and pose.

Some physical constraints of the human body can be modeled in the 3D articulated representation. For example, certain joints can only vary in a limited range of angles, such as an elbow joint can only vary between 0° and 180° or the body parts cannot penetrate into each other. Considering such physical constraints in kinematic representation of human subject have both advantages and disadvantages depending on the problem domain. On one hand, these constraints restrict the admissible search space and prune some unrealistic solutions. On the other hand, it is not easy to handle these constraints in the continuous optimization methods used in generative models.

Diversity and lack of structure in human behavior also add to the complexity of the problem. Although, some human motions have a repetitive structure, like walking or running, others have various levels of complexities, such as common gestures during discussion. Nevertheless, if such routines and structures can be identified, they will be very helpful in tracking and reconstruction. However, all human activities are not structured and these activities are difficult to recover from images.

1.3.2 Appearance and Occlusion Effects

Loose clothing that deforms is also a source of ambiguity. Current tracking algorithms assume fairly tight clothing which reveal the human limbs. A robe or a long skirt which occlude the pose of the limb entirely are not usually considered. Indeed, strong variability in appearance and shape is difficult to model.

In addition, identifying the features belonging to background and foreground is another vision difficulty. In certain situations, some background objects look similar to individual limbs, such as trees, edges, ridges or pillars. All possibly available information must be used to distinguish between human and non-human structure. Self-occlusion between different body parts occur due to highly articulated body structure. It is important to detect and predict self-occlusion as it results in singularities [119]. In this regard, learning a priori knowledge plays an important role in filling the missing data.

Variations in lighting conditions also affect the appearance based image features such as edges or color intensities. Cast shadows also produce artificial edges. Lighting variations complicate texture modeling. Similarly, if a fast motion is captured by a slow shutter camera, motion blur occurs. This will also affect the internal appearance structure, which will be blurred with respect to the orientation of motion.

1.3.3 Kinematic/Observational Singularities

The kinematics of an articulated object provide the most fundamental constraint on its motion. It is required that the state space must be fully observable [119]. Loss of observability occurs when some states have no instantaneous effect on the image features. In this situation, the kinematic Jacobian loses rank and becomes singular which results in numerical instability and leads to tracking failure. This singularity arises physically when the link rotates through the plane parallel to the image plane, resulting in a point velocity in the direction of camera axis.

Ambiguities also arise when some of the model states cannot be directly inferred from the image observations. It depends upon the design of image features and observation models. It is required to solve self-occlusion issues to avoid observation singularities. As an example, when a human arm is straight and edge-based observation likelihood with a symmetric body part model is used, rotation around the limbs own axis cannot be observed. The occluding contour changes little when the limb rotates. The rotation will be visible only when the arm bends at the elbow joint. If an intensity model is used, this ambiguity will not occur.

1.4 Research Challenges

Pursuing the proposed research direction encounters a variety of challenges. Sample based sequential Monte Carlo (MC) techniques, such as PF, were successfully applied to a number of state estimation problems but they are computationally more expensive than Kalman Filter (KF). In visual tracking, the discrete observations $\boldsymbol{z}_t$ which are used for likelihood computation are not directly available. For example, there is no simple detector that can give a contour or a silhouette as a measurement, if it is to be used for tracking. Therefore, low level image processing is required to collect the image features that will be later used for likelihood computation. For a large number of particles, evaluation of likelihood becomes computationally very expensive.

For three-dimensional articulated motion tracking, a coarse body model requires 25 dimensions. Therefore, the state variables are tracked in a high dimensional space. Moreover, the search space is usually complex and multimodal. This high dimensionality complicates the problem further as the number of particles required for tracking grows exponentially in order to cover the regions of state space where true states possibly reside. Thus, it is required to effectively and efficiently draw samples to restrict the sampling volume to the 'more likely' regions.

Various approaches, such as sample smooth method [49] and local MC methods, such as local linearization [37], auxiliary PF [105], kernel smoothing [22], Markov Chain Monte Carlo (MCMC) [109], were proposed to alleviate these issues. Recently more methods, such as multiple hypothesis tracking [142], partitioned sampling [87], covariance scaled sampling [138], Annealed Particle Filter (APF) [36] have demonstrated efficiency over the generic PF. Some of these techniques [142, 87, 138, 36] use complicated optimization methods to allocate the particles in 'important' areas of the search space. Although these approaches have reduced the requirements of the number of particles and have obtained satisfactory estimates but they also make sacrifices. For example, the relocated particles are not fair samples from the posterior distribution anymore. The result is that the estimates obtained may not be strictly minimum Mean Square Error (MSE) estimates.

1.4.1 Problem Statement and Research Goals

Generic PF approaches suffer due to the 'curse of dimensionality' and the challenge of 'particle degeneracy'. The requirement of a high number of particles for reliable tracking adds to the computational cost. Degeneracy occurs when only a few particles have significant weights. As a result, the particles after resampling do not remain true representatives of the posterior distribution. Thus, generic PF loses the diversity of exploring the whole solution space to find the global solution. As a result, the solution obtained may not be a global minima or the solution may even diverge.

Our research goals are motivated by the challenges these problems pose to AHT and are as follows:

- To apply swarm intelligence and evolutionary computational techniques to find the most likelihood region in the search space. The result of using population based optimization algorithms is the reduction in the number of particles required for tracking. These methods have proved successful in various optimization fields. We deploy those techniques to address the requirements of AHT.

- To develop efficient swarm behavior in the proposed optimization algorithms to address the problem of 'degeneracy'.
- To develop an effective hybrid optimization method for AHT by benefitting from the specific advantages of different optimization approaches by combining them.
- To investigate the proper balance between exploration and exploitation of the solution space in swarm and evolutionary algorithms in AHT problem domain.

1.4.2 Research Contributions

The achievement of these goals has resulted in the following research contributions:

- An easy to understand and render schematic of the probabilistic generative whole body tracking approach is suggested. This intuitive scheme has delineated the tracking framework at a higher-level of abstraction.
- Four novel algorithms based on swarm intelligence and evolutionary computation are developed and tested.
- The proposed algorithms demonstrated reduced particle requirement for tracking which improves the computational efficiency. Hence, the proposed methods are more viable for real time applications.
- The proposed algorithms resolved the 'degeneracy' problem of the generic PF as the whole swarm moves towards the optimal solution by searching a wider solution space and sharing information.
- An adaptive parameter update procedure based on the fitness of the solution is proposed which has imparted an automatic ability to switch between exploitation and exploration.
- The decentralized search structure of the proposed algorithms allow them to recover from catastrophic failures.

- The use of evolutionary search algorithms has alleviated the requirement for an accurate dynamic model as the optimization algorithm is able to search more likelihood image regions via the embedded diversified search mechanism.

- The proposed algorithms successfully demonstrated improved performance on parameter identification, low-dimensional tracking and high-dimensional AHT problems.

1.5 Thesis Organization

The remainder of this dissertation is organized as follows. *Chapter 2* reviews the relevant literature on AHT. *Chapter 3* delineates the tracking methodology and describes the video and motion capture dataset intended to be used in this research work. It gives an overview of the overall tracking framework and describes its various components. *Chapter 4* proposes a new Adaptive Parameter Particle Swarm Optimization (AP-PSO) algorithm for AHT. *Chapter 5* presents a Modified Artificial Bat (MAB) algorithm for AHT and *Chapter 6* gives Differential Mutated Artificial Immune System (DM-AIS) algorithm for AHT. *Chapter 7* describes hybrid Particle Swarm Accelerated Artificial Immune System (PSO-AIS) algorithm developed in this research. Preliminary experimental results of parameter identification and low-dimensional tracking of the developed algorithms are given in their respective chapters. *Chapter 8* presents a comprehensive performance analysis of the proposed algorithms in terms of scalability, accuracy, precision, recovery and time complexity. Finally, *Chapter 9* summaries this dissertation with conclusions and also reveals some future research directions.

Chapter 2

Literature Review

AHT is a challenging computer vision problem. This chapter provides the literature review of the stat-of-the-art in the field of human tracking. Section 2.1 describes the application areas, main issues and a broader classification of methods used for AHT. Section 2.2 illustrates the two approaches adapted for three-dimensional AHT. Section 2.3 investigates the research work related to the generative approach of AHT.

2.1 Introduction

A large body of research literature addresses the topic of AHT and human pose estimation as well as the related topics of people detection. The applications include robotics, human computer interface, gesture recognition, biomechanics, image content extraction, surveillance, and image analysis for computer animation. Each application has different requirements in terms of posture complexity, desired output, image complexity and view changes. Existing algorithms are computationally expensive due to the large number of degrees of freedom associated with the human body and the ambiguous nature of a 2D projection of a 3D scene. Several survey papers [59, 44, 90, 91] provide reviews of the earlier work in the addressed areas, but in recent years quite a few other techniques have been proposed.

There are two main issues related to 3D motion based AHT. Firstly, the high dimensionality of the human body, a coarse description of human body requires 25-degrees of freedom. Secondly, there is ambiguity in the information obtained from videos due to self-occlusion, and baggy clothing. As far as self-occlusion is concerned, use of multiple-cameras situated at suitable places can reduce this issue. The high number of degrees of freedom issue can be alleviated either by localizing the limbs of the human body independently or by constraining the possible configuration of the human body, such as running, walking and dancing.

AHT and pose estimation systems vary in the degree of complexity and can be broadly divided into image feature based systems, 2D human model based systems and 3D human model based systems. Considering the scope of this research, the literature review is only focused on 3D model-based tracking of human motion.

2.2 3D Human Model-Based Systems

The approaches for 3D articulated human tracking can be divided into Generative and Discriminative approaches. The generative approach is a bottom-up approach which utilizes some sort of human model, whereas, the discriminative approach is a learning-based approach, used to construct a classifier to detect articulated motion.

2.2.1 Generative Approaches

Generative approaches try to model the image formation process. In generative approaches of AHT, a 3D articulated model is used. The pose is defined by a set of parameters representing the global position and orientation of the root joint (mostly torso) and the joint angles that each limb makes with respect to the limb higher up in the tree. The pose is projected onto the image plane and an error function is computed to find the mismatch. The model parameters are then updated to minimize this mismatch. The method requires the model configuration of the previous frame and later on it is only required to update the parameters for the current frame. This requires model initialization at first frame which is an open problem. To date, most models are initialized manually. The use of kinematics and 3D body shape models in a generative framework is the most popular approach for human pose estimation and is adopted in this research study as well.

2.2.2 Discriminative Approaches

In discriminative approaches, the problem is formulated as a classification problem. It is more complicated to estimate the underlying kinematics or the skeletal articulation structure of a person as done in generative approaches. In the discriminative approach a classifier or regressor is learned from a set of body features obtained for a particular part. The input image features are fed to the classifier or regressor to directly reconstruct the 3D pose.

The mapping from image observations to 3D pose is generally ambiguous since different poses can generate the same silhouette, resulting in a one-to-many problem. Another issue is the requirement of a huge amount of data for classifier learning which may not be readily available. A change in a motion model, say from walking to jumping, require classifier learning to the new motion type.

2.3 Generative Approaches: State-of-the-Art

This research work is related to a 3D model-based tracking of a human subject so the literature review is mainly focused on generative approaches.

2.3.1 Use of Geometric Primitives for Appearance Modeling

Generative approaches have been used since the mid 1980's. Hogg et al. [53] used 3D geometrical models with body kinematics, an observation model and a method to estimate the 3D parameters based on the matching of the projected model on the image plane. They used a hierarchical model with parts represented by cylinders. The use of other image primitives such as cylinders, cones and ellipsoids [104], deformable models [63] and implicit surfaces [107] are also reported in AHT studies.

Sidenbladh et al. [129] carried out the articulated tracking by modeling the human body using articulated cylinders as body parts. The appearance model is constructed by projecting each body part into a reference image [128]. This 3D articulated appearance model is then employed for tracking using a PF framework. This technique works well for single body part tracking but only works for specific motion models (walking, running) for whole body tracking.

Delamarre et al. [33] used geometric primitives such as cylinders and cones for articulated model construction. The articulated model is constrained from the multi-view silhouettes. Geometric constraints are used to generate the 'forces' to align the 2D contours of the projected models with the silhouette boundary.

2.3.2 Use of Dynamic Models for Motion Prediction

Use of a dynamical model to predict the motion in consecutive frames and its integration with image observations to form a filtering framework is extensively employed in AHT research. Mikic et al. [88] used KF and Sidenbladh et al. [130] used PF to predict the motion dynamics. Human body dynamics are modeled differently in different studies. Kakadiaris et al. [63] used a constant velocity or constant acceleration model, Pavlovic et al. [103] used a switching linear dynamical model, and Sidenbladh et al. [130] used an example based model (learned model). Sidenbladh et al. learned and used a low dimensional linear model of human motion to structure the example motion database into a binary tree. An approximate probabilistic tree search method exploits the coefficients of this low-dimensional representation. This probabilistic tree search returns a particular sample human motion with probability approximating the true distribution of human motions in the database. This sampling method is suitable for use with PF techniques and is applied to articulated 3D tracking of humans within a Bayesian framework. Plankers et al. [106] used a deterministic gradient descent-based approach to iteratively estimate pose changes and this estimate is updated after each step. The study was successful in tracking arm movement with self-occlusion using the silhouette.

Gradient-based methods of object tracking [17, 18, 34], tend to loose track of the limbs due to the complexity of the articulated human model. The sampling-based (statistical) approaches [24, 35, 138], can overcome some degree of pose ambiguity and can track for longer durations by representing the pose estimation with a set of samples. Inverse kinematics are also employed for AHT. The problem with inverse kinematics is that many possible inverse kinematics exist for a single image. Picking a wrong solution will result in tracking failure. Multiple hypothesis techniques, such as PF, could give better performance in this situation. A PF estimates the dynamical state with ambiguous observations by approximating the state posterior distributions [86]. However, due to the high dimensionality of the human model, a sufficiently large number of particles is required, otherwise the PF is likely to degenerate to characterize a complex state posterior distribution. MacCormick et al. [87] used a partitioned sampling approach of the state space for 2D tracking but the approach cannot be extended directly for 3D whole-body pose estimation due to the increased dimensionality of the model. Deutscher et al. [35] proposed APF to combine a deterministic annealing approach to reduce the number of required samples. The system is successful in pose tracking with prominent observations. However, the method may fail due to the self-occlusion and disappearance/reappearance of body parts. Urtasun et al. [157] presented a method to 'relocate' the particle-depleted regions of state space. Sminchisescu et al. [136] assumed a low dimensional dynamic space by constraining to a certain type of movement, say walking or running. But the method will be less applicable for uncontrolled human motion. Schmidt et al. [124] used a kernel PF for the tracking of 3D upper body.

Sminchisescu et al. [138] proposed a sophisticated covariance scaled sampling and search strategy in a 30-dimensional state space to overcome the depth ambiguity of the joints. The method operates by incorporating the domain knowledge of the 3D human structure into the sampling scheme. A cost function is used including multiple cues such as intensity and optical flow for human tracking. The covariance is increased in the direction of greatest uncertainty to make the probability of sample generation close to minimum. A steepest descent approach is then used to optimize the sample to find local minima. The method is further improved with the use of a mixture smoother in a later work [136]. A smoother gives an estimate of the state of the system at some time-step given all the measurements including ones encountered after that particular time-step. The used layered mixture density smoother, exploits the accuracy of efficient optimization within a Bayesian framework. The distribution is progressively refined by combining polynomial time search over the embedded network of temporal observation likelihood peaks, maximum a posteriori continuous trajectory estimates, and Bayesian variational adjustment of the resulting joint mixture approximation. They further explored the use of potential kinematic minima that causes the visual ambiguities to improve the sampling efficiency which gives robust reconstruction of human motion in monocular videos [139].

Lee et al. [75] used a probabilistic proposal map along with 3D model to estimate the likelihood of body parts in different 3D locations to recover the 3D body pose. Each image position is evaluated to get the hypotheses about the body parts. These hypotheses are then compared with the image observations and the proposal maps of various body parts are generated. This proposal map gives the probability of the existence of a body part on different image locations. A data driven MCMC approach is used for searching the pose space. Navaratnam et al. [96] detected different body parts in each frame of the monocular sequence. The 3D pose is recovered using a hierarchical kinematic constraint between the body parts. They used a Hidden Markov Model (HMM) framework to integrate the temporal information which is used for temporal coherence. One important issue with sample-based methods is the initialization of the pose which is done manually.

Bullock et al. [19] presented a complete visual tracking system where target detection, target model acquisition/initialization and target tracking components are included in a single probabilistic framework. Visual cues are used to detect the target, physical dimensions of limbs are learnt, color data with spatial constraints is used for appearance and PF is used for tracking.

Samantha et al. [97] used coronal (frontal) plane images to detect walking motion of rollator users. Only a monocular view of the lower limbs and a frontal perspective of the rollator user is employed. They used multiple cues using three image features: color, image gradient and anthropometric symmetry, for pose estimation and later used Bayesian probabilistic framework for posterior estimation.

Yao et al. [169] presented an efficient stochastic gradient descent algorithm that learns probabilistic non-linear latent spaces composed of multiple activities. They further derived an incremental algorithm for online update of the latent space without extensive relearning.

Zhang et al. [173] proposed a pose search method by introducing a new generative sampling algorithm with a refinement step of local optimization. They introduced a novel way of handling the physical constraints and employed 3D distance transform for speedup. They developed optimized parallel implementation on Graphical Processing Unit (GPU) and attained full body tracking speed of 9 fps.

Saini et al. [122] studied the use of stochastic filtering, such as PF and APF and stated that these algorithms are computationally very costly and are unable to produce good accuracy. In contrast, stochastic evolutionary optimization algorithms, such as Particle Swarm Optimization (PSO), are more accurate but some times suffer due to the existence of several local minima. They proposed quantum-inspired PSO algorithm that has fast convergence ability and can escape local minima.

2.3.3 Use of Graphical Methods for Pose Estimation

Sigal et al. [134] presented a Non-parametric Belief Propagation (NBP) method for pose estimation. NBP is an inference algorithm for graphical models containing continuous, non-Gaussian random variables. Like PF, NBP approximates the posterior distribution as a collection of samples. However, NBP uses a graphical structure to greatly reduce the dimensionality of these distributions. A belief network of each node corresponding to parameter representing body parts is constructed. Belief on each component is propagated along its network and combined with inferences from other components to estimate the pose. The technique requires the formation of joint potential functions of various nodes in the network and the associated conditional distributions of various observations. The method is computationally more efficient than sampling methods such as MCMC as it requires less iterations.

NBP has limitations with regard to global constraints. The structure of the belief network resembles the articulated human body structure but only the constraints between adjacent components are modeled. Non-self penetration constraints, which ensures that the body parts don't penetrate into each other, is not considered. As a result, an invalid pose is not penalized in pose estimation.

Disaggregated models can also be formulated using undirected graphical models. If the body parts are assumed independent then the human body can be modeled using undirected graphical model and can formulate tracking as pose estimation. Felzenszwalb et al. [39] developed a linear complexity exact inference scheme using such graphical models and standard belief propagation. It is used for 2D pose estimation. A tree structured topology for the graph, a particular form of potential function, which encodes connectivity of the body parts as joints, and discretization of the state space is used. As a result the efficiency comes at the cost of expressiveness. Another drawback is that the model cannot account for occlusion, temporal constraints or long range correlation between body parts. The method is also impractical for 3D inference. These models are later extended by Lan et al. [72] for 3D body tracking by adding correlation between body parts and Ramanan et al. [112] by jointly learned appearance.

More recently, Bergtholdt et al. [14] introduced a denser connected graph. They have used a *-search to find globally optimal solution which is lower-bounded by an admissible heuristic. It deals with the long range correlation but used for simple 2D kinematic structure. On similar lines, Tian et al. [150] introduced branch-and-bound approach by using a true structured solution as a lower bound. Sigal et al. [133] provided a scalable solution with a trade-off between expressiveness and computational resources. Their method permits expressiveness similar to [39] and has no assumption about the topology of the graph and allows for a richer class of potential functions. It can produce a continuous estimation of the pose in 3D. They used a variant of NBP for approximate inference.

Daubney et al. [30] presented a method that modeled greater uncertainty over the root node without increasing the uncertainty of the remaining parts of the model, making it less vulnerable to tracking failure. All probability distributions are approximated using a single Gaussian allowing inference to be carried out in close form. In a later work [31], they used a sparse set of moving features for articulated pose estimation. The method is entirely dependent on motion of regions rather than appearance. A standard feature tracker is used to automatically extract a sparse set of features for tracking. The reported quantitative results showed improved performance over the state-of-the-art.

2.3.4 Use of Learned Human Motion Models

Recently, learned models of human pose and motion are used to constrain the reconstruction of human movement from single or multiple views. Sidenbladh et al. [127, 129, 130] learned dynamic priors from training data obtained from motion capture. They used an indexed database of motion capture examples to obtain the direction of motion. Thayananthan et al. [148] learned statistical priors of human appearance, edges, ridges and image motion. These statistical priors are used as various image cues for a given movement. A tree-based representation of hand dynamics is built. Each tree node corresponds to one partitioning of the state space with piecewise constant densities. The tree is constructed in two ways. In the first method, a hierarchical clustering algorithm is applied to align the collected data points with the cluster centers which act as nodes. In the second method, a lower dimensional eigen space is partitioned using a grid at multiple resolutions. In this, each partition center corresponds to a node in the tree.

Agarwal et al. [1] learned second order dynamics for 2D tracking of walking and running with transitions and turns in monocular image sequences. Urtasun et al. [157] used Principal Component Analysis (PCA) to provide a low dimensional parametrization of concatenated joint angles from multiple examples of walking and running sequences. Sigal et al. [131] presented an automatic 3D pose initialization method. A learned motion model's information is combined with a body parts detector for tracking walking people. Rosenhahn et al. [118] presented a markerless human motion capture method to learn scaled motion dynamics. The motion is modeled by twists which allow proper scaling of the prior. It eliminates the need of training data for different frame rates and velocities. This method can combine many different motion patterns.

2.3.5 Use of Dimensionality Reduction Techniques

Recently, use of non-linear probabilistic generative models is reported in literature. Gaussian Process Latent Variable Model (GPLVM) is a dimensionality reduction technique. It is a probabilistic non-linear PCA. It uses a kernel function to form the mapping in the form of a Gaussian process. In [50, 73, 155], GPLVM is used to model the low dimensional full body joints, and in [149] it is used for upper body joints, in a probabilistic framework. Grochow et al. [50] learned Scaled Gaussian Process Latent Variable Models (SGPLVM) to model the probability distribution of whole body poses. Urtasun et al. [155] developed the Wandering-Stable-Lost (*WSL*) tracker for tracking human joints. It is a three component tracker: the first component W accounts for the short term changes to the appearance model, the second component S is related to the stable image structure and the third component L accounts for the data outliers. The authors are successful in obtaining an optimized model joint reprojection error of the model in a low dimensional space using SGPLVM. GPLVM is a powerful static modeling approach. It does not provide a smooth data for time series human movement poses. Gaussian Process Dynamical Models

(GPDM) [160, 94] and balanced-GPDM [156] are variants of GPLVM. They have proven to be powerful enough to capture the underlying dynamics of movement as well as reducing the dimensionality of the pose space. Such models have been successfully used as priors for kinematic tracking of walking [156].

Manifold learning is a powerful tool for dimensionality reduction. It is a non-linear, geometrically intuitive and computationally feasible dimensionality reduction technique. It is based on the assumption that the input data lies on or close to a smooth low-dimensional manifold. Manifold learning algorithms attempt to preserve a different geometrical property of the underlying manifold. Li et al. [77] used a mixture of factor analyzers to locally approximate the pose manifold. Factor analyzers perform non-linear dimension reduction and data clustering concurrently with a global coordinate system. Factor analyzers have made it possible to derive an efficient multiple hypothesis tracking algorithm based on distribution modes.

Taylor et al. [146] presented a model based on the Conditional Restricted Boltzmann Machine (CRBM) that preserves simple exact inference property and includes multiplicative three-way interaction that allows the effective interaction weight between two units to be modulated by the dynamic state of a third unit. They factorize the three-way weight tensor implied by the multiplicative model. This efficient compact model is then used for modelling human motion.

Li et al. [78] used a globally coordinated mixture of factor analyzers that is learned from motion capture data to approximate the low-dimensional manifold embedded in the high-dimensional human pose space. Each factor analyzer in the mixture is a 'locally linear dimensionality reducer' that approximates a part of the manifold. These locally linear pieces are then aligned to get the global parametrization of the manifold. They also proposed a variational Bayesian formulation to enable automatic and optimal selection of factor analyzers and dimensionality of the manifold.

Raskin et al. [113] proposed Hierarchical GPLVM (H-GPLVM) to create a hierarchy of manifolds trained on two different activities. The hierarchical approaches extend the pose space by decoupling the motion of individual body parts. In a recent paper [114], they combined GPLVM and APF to produce Gaussian Process Annealed Particle Filter (GPAPF) method that constrained the solution space. Human body motion is described by concatenation of low-dimensional manifolds that characterize different motion types. They introduce a body pose tracker that uses the learned mapping function from latent space to body pose space. The trajectories in the latent space provide low dimensional representations of body pose sequences representing a specific action type. These trajectories are used to classify human actions.

Moutzouris et al. [95] proposed a Hierarchical Temporal Laplacian Eignemaps (HTLE) for leaning their action manifold using training set from MoCap data describing the action of interest. Image captured by multiple calibrated synchronized cameras are used for visual hull estimation. The visual hull and action manifolds are used by hierarchical manifold search algorithm for 3D pose estimation.

2.3.6 Use of Autonomous Initialization of Tracker

Integration of generative and discriminative approaches for articulated human motion tracking are also reported. Curio et al. [25] proposed a 2D articulated motion tracking system. A one-to-many Support Vector Regression (SVR) system is learned to conduct view-based recognition. State dynamics are represented by a first order AutoRegressive (AR) model. A competitive PF defined over the hidden state space is deployed to select plausible branches and propagate state posteriors over time. This system has demonstrated autonomous initialization due to SVR. It draws samples from both current observations and state dynamics. Only the best body configuration from the view-based and the model-based matching is used to initialize the tracking.

Using a single initial state runs the risk of missing other admissible solutions due to the inherent ambiguity. Sminchisescu et al. [137] presented a joint learning algorithm for a bidirectional model that combines the generative and discriminative processing. The model was successfully used for 2D people detection and 3D human motion reconstruction from static images with cluttered background. However, parameter learning of the bidirectional model is only considered, no movement dynamics are used.

More recently, there have been attempts at building generative approaches that are able to initialize automatically [40, 41, 60]. John et al. [60] used a hierarchical strategy along with an efficient evolutionary search algorithm to estimate the pose of the body in the first frame of a sequence. The hierarchical search assumes that the segments higher in the kinematic hierarchy can be localized well to reduce the search for subsequent body parts. The method partitions a high dimensional search into a number of smaller conditional searches. Gall et al. [40, 41] proposed a framework based on simulated annealing that allows automatic initialization by directly searching for the global optimum of the objective function. The results are compelling and the approach is only guaranteed to converge to the global optimum. The method also has the ability to localize the body in space in terms of a rough bounding box, which is not easy in noisy scenarios.

Salzmann et al. [123] presented a unified framework for articulated pose estimation and non-rigid shape recovery. Their introduction of distance constraints into discriminative method has improved its performance. Later on, generative method is used to prevent the drifting of the proposed pose from image observations. They also studied different relaxations of distance constraints.

Sedai et al. [125] proposed a hybrid method that combines Gaussian process learning, a PF and annealing to track the 3D pose in video sequences. Their approach comprises of two steps: training and tracking. In training step, a supervised learning method is used to train a Gaussian process regressor which takes the silhouette features as input and produces (estimates) multiple output poses modelled by a mixture of Gaussian distributions. In the tracking step, the estimated poses of the Gaussian process regression is refined by APF to track the 3D pose. They claimed that their method does not lose tracking.

Liu et al [83] proposed an improved silhouette extraction method. Shadows are detected and removed and a level set method is employed to achieve better silhouette extraction. Further, they proposed an adaptive view fusion approach for improved matching of human 3D model and observations.

Chapter 3

Tracking Methodology

This chapter delineates the experimental set-up for articulated human tracking. The basic components of the set-up are explained in details along with the ground truth and video data sets. Section 3.1 provides some insights into the generative model-based object tracking framework. Section 3.2 describes the temporal model which is used to obtain the state prior. Section 3.3 describes the hierarchical human model used in this study. Section 3.4 talks about image features and the computation of the image likelihoods related to this study. Section 3.5 explains the ground truth and video datasets utilized in this study.

3.1 Tracking Framework

Generative method consists of four basic components: initialization (defining the human model parameters), a state prior (pose hypothesis: temporal model), an observation likelihood model (matching function) and a search algorithm. These components are delineated in Figure 3.1.

Figure 3.2 shows a detailed block diagram of the generative model based Bayesian approach for AHT. At the beginning of a video sequence, a 3D articulated model of the human body is initialized manually using ground truth data. It is assumed that the posterior estimates of the states at the previous time step are available. A dynamical model is then used to advance the state estimates to the current time step. The angular positions of specific joints are the model states. In a size N PF framework, each particle represents a specific subject pose as a set of joint angles. The 3D human model joints are then positioned according to these predicted joint angles to get the pose proposed by each particle. Since

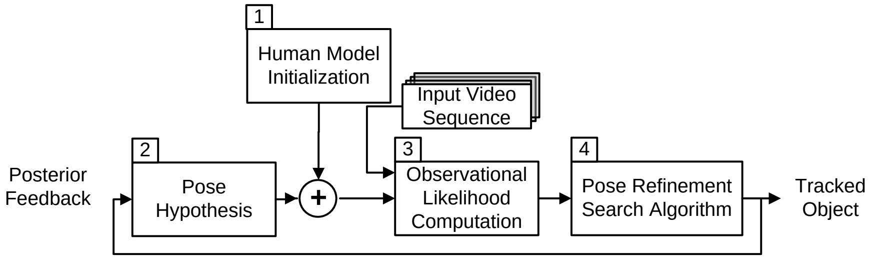

1
Human Model Initialization
Input Video Sequence
2
Posterior Feedback
Pose Hypothesis
+
3
Observational Likelihood Computation
4
Pose Refinement Search Algorithm
Tracked Object

Figure 3.1: Generative methodology: basic components

N-particles are employed, N pose proposals are obtained. These poses are projected onto the image plane and pose likelihoods are computed. These likelihoods estimate how well the projected poses fit to the image observations using edge and silhouette likelihoods. On the basis of these likelihood functions new weights are assigned to each particle. Then the 3D human model joints are positioned according to the average/best joint position obtained from these updated weights, Image Likelihoods (ILH), to get the posterior pose.

In case of a generic PF, the dotted block (metaheuristic optimization algorithm block) is bypassed. Resampling step is carried out if degeneracy condition is detected [86]. In case of metaheuristic optimization augmented PF, the N-estimates will act as the initial generation of the metaheuristic optimization block and their likelihoods will act as the fitness value of each particle. The swarm intelligence or evolutionary computation based optimization algorithms then iteratively operate and move the swarm towards the high likelihood regions. The particle weights are then updated and normalized and final pose is estimated. The particles are then used for next time step temporal prediction.

3.2 State Prior: Pose Hypothesis

Understanding of human actions and intensions by vision is an essential and challenging computer vision problem. The objective is to estimate the trajectory of human limbs based on image data. Ideally, a tracking algorithm should be able to locate the object position anywhere within the given image. However, due to computational efficiency reasons, only a limited region of the image is searched [91]. As a result, the object can only be tracked if it does not move beyond the search region. To avoid this problem, the target location is first estimated and the search is carried out in a region centered at this estimated

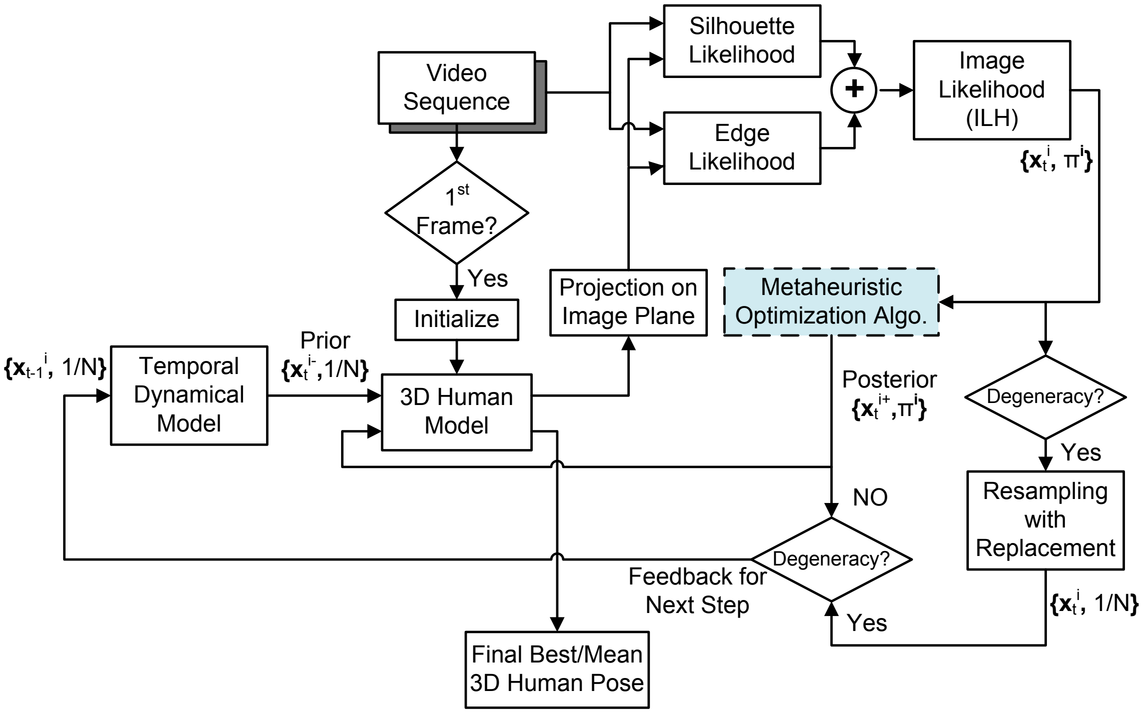


Figure 3.2: Monte Carlo approach for articulated object tracking using metaheuristic optimization

position. The estimation process involves two components: the physical measurements provided by available sensors and knowledge of the process dynamics. KF [64] and its variants, Extended KF (EKF) and Unscented KF (UKF), have been successfully applied on a large number of state estimation problems [135, 110]. EKF linearizes the nonlinear state transition and observation models around the current estimates. These linearized models are then used in the KF equations [135]. UKF uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points, called sigma points, around the mean. These sigma points are then propagated through the nonlinear state transition and observation models, and the covariance of the estimate is then recovered [135].

In case of highly non-linear and non-Gaussian models, the KF based methods which depend on linearization, are often inadequate. PF is a very effective estimation framework in this situation [111]. PF is a probability based estimator which is based on Bayes' rule. PF does not make any assumptions about the shape of the error distribution, but instead estimates this by a population of sample particles.

Traditionally, a zero velocity or constant velocity [12, 36, 128] temporal model is used for pose prediction for the next time step. Recently, discriminative matching functions such as GPLVM [50, 73, 155], HTLE [95], Gaussian process learning [125], H-GPLVM [113] are used to generate the pose hypothesis space. The final pose is then estimated by searching the hypothesis space using a global or local search algorithm.

3.2.1 Pose Hypothesis and Refinement using Generic PF

In this research, PF, which supports multiple hypothesis, is used to estimate the target position for the next time step and later on metaheuristic optimization algorithms are used to refine this predicted target position. Hence, the PF is the basis of this tracking framework.

A generic PF algorithm consists of four steps: initialization, prediction, update and resampling. During the *Initialization* step, N-samples are drawn from the initial i.i.d. (independent and identically distributed) distribution. Afterward, the values of the particles for the next time step are *Predicted* according to the dynamics of the process. During the *Update* step, each predicted particle is weighted by the likelihood function, which is determined by comparing the predictions with the observation. In the Resampling step, the particles are selected on the basis of normalized weight function (see Appendix B, Equation B.23). It gives birth to new particles at the expense of less likely particles.

Figure 3.3 represents a PF framework for a one-dimensional (1D) signal. Gray circles represent the unweighted particles and black circles represent the weighted particles. Their horizontal positions indicate their values in state space. The 'size' of each particle is relative to its weight. Larger particles will produce more offsprings during the resampling step.

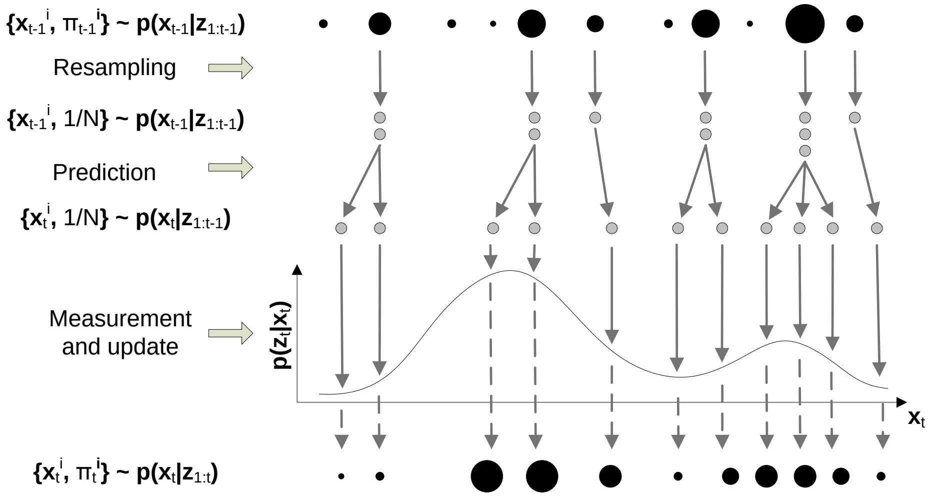

$\{x_{t-1}^i, \pi_{t-1}^i\} \sim p(x_{t-1}|z_{1:t-1})$
Resampling
$\{x_{t-1}^i, 1/N\} \sim p(x_{t-1}|z_{1:t-1})$
Prediction
$\{x_t^i, 1/N\} \sim p(x_t|z_{1:t-1})$
Measurement and update
$p(z_t|x_t)$
x_t
$\{x_t^i, \pi_t^i\} \sim p(x_t|z_{1:t})$

Figure 3.3: A symbolic representation of PF framework

3.3 The Articulated Human Body Model

In this work, the human body is modeled as a set of 10 cylinders of different sizes connected by joints [129],[12]. This is called an articulated body representation. The body parts included are the head, torso, upper arms, forearms, thighs and calves. The torso is the base of the articulated model and all other parts are linked to it through revolute joints.

A revolute joint has 1-degree of freedom. Its action causes a pure rotation of one cylinder with respect to the other. If a body part has more than one degrees of freedom, for example the shoulder has 3-degrees of freedom, then it is constructed by joining 3-revolute joints with all their axis intersecting at one point. This makes it clear that the basic unit of the human motion is simply a rotation, that is all human actions are combinations of rotations of the body parts around their respective joints.

A kinematic hierarchical structure is used to represent the human body. Each body

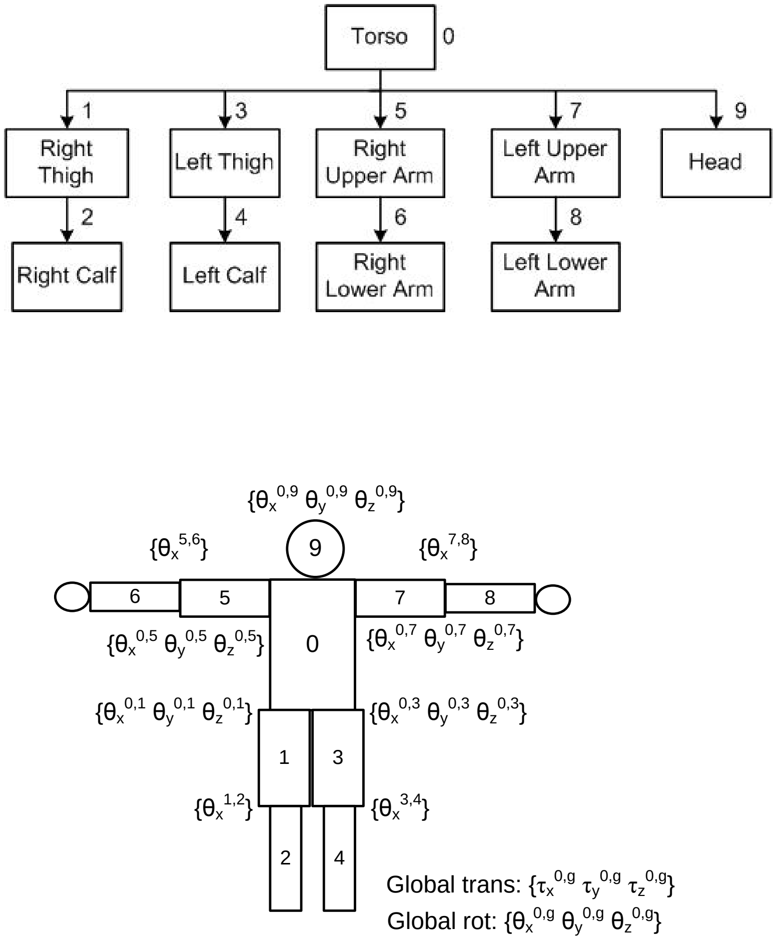

Torso
0
1
3
5
7
9
Right Thigh
Left Thigh
Right Upper Arm
Left Upper Arm
Head
2
4
6
8
Right Calf
Left Calf
Right Lower Arm
Left Lower Arm
$\{\theta_x^{0,9}\ \theta_y^{0,9}\ \theta_z^{0,9}\}$
$\{\theta_x^{5,6}\}$
9
$\{\theta_x^{7,8}\}$
6
5
7
8
$\{\theta_x^{0,5}\ \theta_y^{0,5}\ \theta_z^{0,5}\}$
0
$\{\theta_x^{0,7}\ \theta_y^{0,7}\ \theta_z^{0,7}\}$
$\{\theta_x^{0,1}\ \theta_y^{0,1}\ \theta_z^{0,1}\}$
$\{\theta_x^{0,3}\ \theta_y^{0,3}\ \theta_z^{0,3}\}$
1
3
$\{\theta_x^{1,2}\}$
$\{\theta_x^{3,4}\}$
2
4
Global trans: $\{\tau_x^{0,g}\ \tau_y^{0,g}\ \tau_z^{0,g}\}$
Global rot: $\{\theta_x^{0,g}\ \theta_y^{0,g}\ \theta_z^{0,g}\}$

Figure 3.4: The human body model [129]

part is assigned a number of degrees of freedom depending on the number of revolute joints attached to it [2, 4, 5]. Moreover, the joint angles are constrained to anatomic joint angle limits, that is the range of angles the human elbow can move is limited, and these limits are taken into account during the tracking process. Hard priors are implemented to exclude implausible joint angles. The neck, shoulder, and hip joints have 3-degrees of freedom, torso and clavicle joints have 2-degrees of freedom, whereas the elbow and knee joints have 1-degree of freedom. Then, there are 6-degrees of freedom for the global position and orientation of the body. This setup results in a 31-dimensional state space representation of the human body as shown in Figure 4.10. The motion of a point in the body part is determined by forward kinematics. The z-axis of the local coordinate system for each limb is directed along the cylinder axis. Rigid transformations are used to specify the relative position and orientation of body parts and to transform to the global coordinate frame. Camera calibration data are used to initialize the translation vector and rotation matrices. They are used to map from the global coordinate system to the camera reference frames [153, 52].

3.4 Image Likelihoods

Every visual tracking system requires an explicit or implicit model for appearance, shape and dynamics of the object of interest. The lack of a suitable model limits the performance of the tracking system. Silhouette information acts as the appearance model and edge gradient acts as the shape model in our experiments. These image characteristics are used to find the ILH as follows:

3.4.1 Silhouette Information

The silhouette of the object of interest is obtained by statistical background subtraction with a Gaussian mixture model. The foreground pixels are set to 1 and the background pixels are set to 0 to form a pixel map as shown in Figure 3.5(a). The likelihood of a pose is then estimated by mapping a number of points on each limb and onto the silhouette pixel map. The mean square error (MSE) between the predicted and observed silhouette values for these points is computed as [36]:

$$MSE_{silhouette} = \frac{1}{N}\sum_{i=1}^{N}(1 - p_i^s)^2 \tag{3.1}$$

where p_i^s are the values of the pixel map at the N-sampling points taken from the interior of the predicted cylinders making up the human skeleton [36]. The silhouette likelihood $LH_{silhouette}$ is then computed as:

$$LH_{silhouette} = e^{-MSE_{silhouette}} \tag{3.2}$$

3.4.2 Edge Gradient

The edges produced by a human subject give a good outline of visible arms, and legs unless the subject is wearing very baggy clothes or the environment is severely cluttered.

Moreover, the edges are less sensitive to color, clothing, texture, and lighting. In this work, a gradient based edge detection, see Appendix A, mask is used [36] . The result is thresholded to eliminate spurious edges and then smoothed with a Gaussian kernel. It is renormalized between 0 and 1 to produce a pixel map as shown in Figure 3.5(b). In this pixel map each pixel is assigned a value related to its proximity to an edge.

A number of points N are selected on the edge of the predicted cylinders making up the body pose. The MSE between the selected points and the value of the edge pixel map p_i^e at this image position is computed as [36]:

$$MSE_{edge} = \frac{1}{N}\sum_{i=1}^{N}(1 - p_i^e)^2 \tag{3.3}$$

The edge likelihood LH_{edge} is then computed as:

$$LH_{edge} = e^{-MSE_{edge}} \tag{3.4}$$

The two likelihoods are combined together to get the ILH as:

$$ILH = \zeta\, LH_{silhouette} + \eta\, LH_{gradient} \tag{3.5}$$

where ζ and η are constants in the range [0-1] such that $\zeta + \eta = 1$. In these experiments equal weighting is given for the two likelihoods $\zeta = \eta = 0.5$.

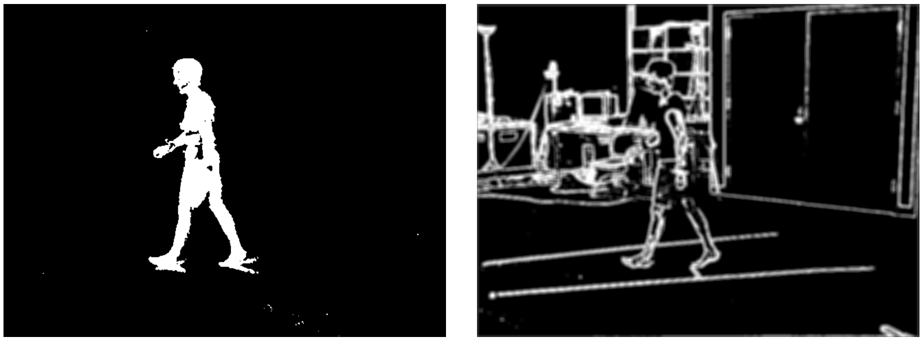
Video sequences obtained from multiple-cameras are employed in this study. The ILH obtained form edge and silhouette likelihoods for each camera are combined together to get and overall $ILH_{all\ cameras}$ function, cost function, as follows:

$$ILH_{all\ cameras} = \sum_{i=1}^{C}(ILH_i) \tag{3.6}$$

where C is the number of cameras used.

3.5 Video Sequences and Ground Truth Data

To evaluate the performance of the proposed metaheuristic algorithms for AHT, a publicly available dataset provided by Sigal et. al. [12, 132] is used. The 3D human motion and video data was simultaneously captured in a laboratory environment. The subjects wore natural clothing, as opposed to tight fitting motion capture suits typically used for motion



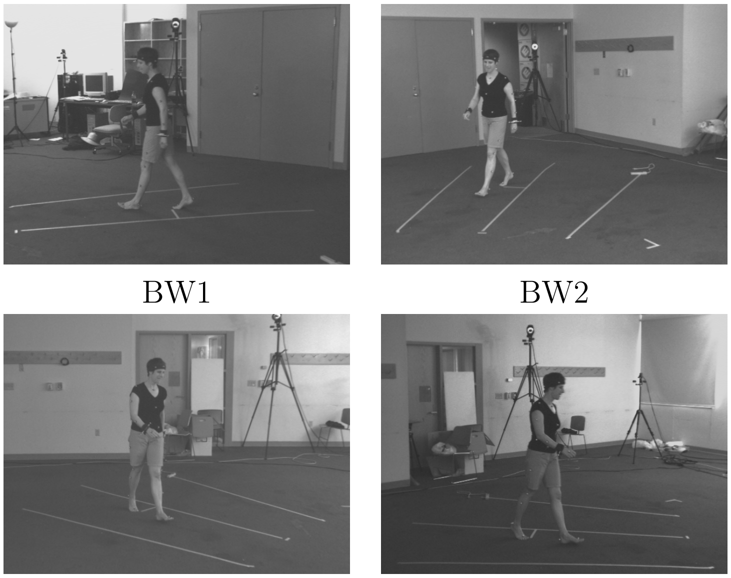
(a) (b)

Figure 3.5: (a) Silhouette (b) Edge pixel map

capture, on which visible markers were attached with adhesive tape. A drawback of this approach is that the markers tend to move more as compared on the tight motion capture suits. As a result, the motion capture data may not be as accurate as obtained by traditional tightly fitted suit wearing objects. Standard markerless motion capture protocols were used, subjects were measured and a 3D body model was fitted to these measurements. The dataset consists of background images, calibration data (camera intrinsic and extrinsic parameters), image data, motion capture data and synchronization data.

For Lee Walk dataset, the ground truth motion data was obtained by a commercial Vicon system [154] that uses reflective markers and six 1 M-pixel cameras. The video was recorded using four Pulnix TM6710 cameras [23]. These were gray scale progressive scan cameras with a resolution of 644 × 488 pixels that record at a frame rate of 60 Hz. The coordinate frames of the motion capture and video capture systems were aligned offline and temporal synchronization was achieved by tracking visual markers on both systems using an optimization procedure. A frame captured by 4-cameras is shown in Figure 3.6.

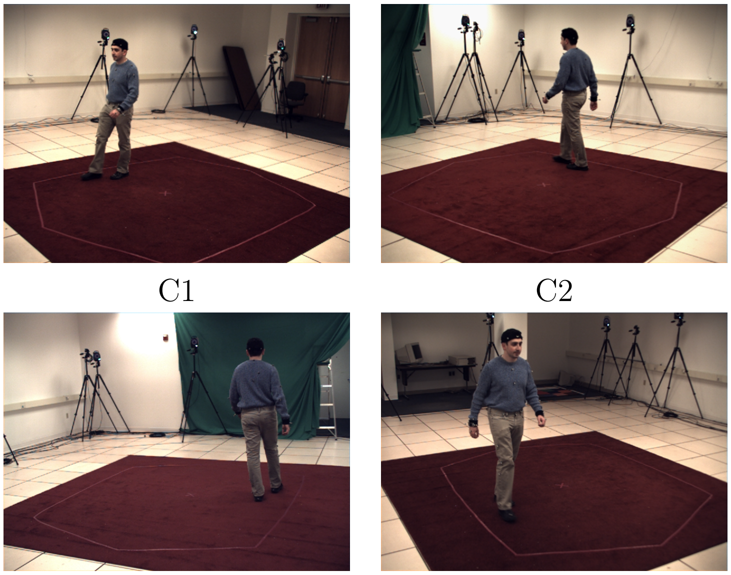
HumanEva II dataset [132] was captured using a more sophisticated hardware system that allowed better quality motion capture data and hardware synchronization. The dataset contains two subjects performing predefined actions. The ground truth motion data was captured by Vicon system which uses twelve 1.3 M-pixel motion capture cameras. IO Industries [54] system that uses four UNiQ UC685CL 10-bit color cameras were used for video recording. It records at a frame rate of up to 110 Hz with 659 × 494 pixels resolution. The raw frames were re-scaled to 640 × 480 pixels using IO Industries software. Videos were captured at 60 Hz. The Vicon system was calibrated using Vicon's proprietary software while the video cameras were calibrated using Matlab camera calibration software. A sample frame captured by these cameras is shown in Figure 3.7.



BW1 BW2

BW3 BW4

Figure 3.6: Lee walk sequence: recorded using 4 gray scale video cameras



C1 C2

C3 C4

Figure 3.7: HumanEva II data set: recorded using 4 color video cameras

Chapter 4

Adaptive Parameter Particle Swarm Optimization Algorithm

In this chapter, swarm intelligence is used to develop a PSO based adaptive parameter optimization algorithm to solve the AHT problem. Section 4.1 gives the rationale behind using swarm intelligence for optimization. Section 4.2 briefly describes the PSO algorithm, which forms the basis of the proposed AP-PSO algorithm. Section 4.3 reviews some of the recent developments to counteract 'degeneracy' challenge in PF. Section 4.4 explains the proposed AP-PSO algorithm. Section 4.5 describes experimental results. Section 4.6 concludes this chapter with a discussion of the obtained results.

4.1 Use of Evolutionary Algorithms

Optimization has countless applications in industry. It results in reduction in cost, time and risk or increase in profit, quality and efficiency. A large number of optimization problems in engineering, science, business and economics are complex in nature. It is difficult to solve them in an exact manner in a reasonable amount of time. Use of approximate methods is a viable solution in such situations. Metaheuristics is a class of such approximate methods. They reduce the effective size of the solution space by exploring it more efficiently. They generally produce faster and robust results in large-scale optimization problems. Swarm intelligence and evolutionary algorithms are two categories of metaheuristics used to solve complex, multi-modal AHT problem in this research.

4.1.1 Problem Formulation

The high dimensional AHT problem is solved as a continuous optimization problem (COP). The model for a constrained COP can be defined as:

$Q = (\mathbf{S}, \mathbf{\Omega}, f)$ of a COP consists of:

- a search space $\mathbf{S}$ defined over a finite set of continuous decision variables and a set $\mathbf{\Omega}$ of constraint relationships among the variables
- a cost function $f : \mathbf{S} \rightarrow \mathbb{R}$

The search space $\mathbf{S}$ is defined as follows: Given a set of continuous variables $\boldsymbol{x}_i, i = 1, \cdots, n$ with possible values $\boldsymbol{v}_i \in \mathbf{D}_i \subseteq \mathbb{R}$. The solution $s \in \mathbf{S}$, a complete assignment where each decision variable has an assigned value that satisfies all the constraints in the set $\mathbf{\Omega}$, is a feasible solution of the given COP. For a minimization problem a solution $s^* \in \mathbf{S}$ is called a global optimum if an only if: $f(s^*) \leq f(s)\ \forall_{s \in \mathbf{S}}$. The set of all globally optimal solutions is denoted by $\mathbf{S}^* \in \mathbf{S}$. Solving a COP requires finding at least one $s^* \in \mathbf{S}$.

The search space for COPs is not finite since the continuous decision variables can assume an infinite number of values. However, solving these problems with digital computers impose certain limitations as computers have a limited degree of accuracy.

Formulation of Video-based AHT Problem

The video-based AHT problem minimizes the image likelihood error from multiple-cameras and/or multiple-likelihoods subject to the joint angle constraints:

$$\begin{aligned}
\underset{\boldsymbol{x_t}}{\text{minimize}} \quad & -\,log\, p(z_t|\boldsymbol{x}_t) = \frac{1}{K}\frac{1}{|L|}\sum_{k=1}^{K}\sum_{l \in L} -log\, p^l(z_t^{(k)}|\boldsymbol{x}_t) \\
\text{subject to} \quad & \boldsymbol{x}_t = \{x_t^1, \cdots, x_t^d, \cdots, x_t^D\} \in \mathbb{R} \\
& x_t^d \geq \theta_{min} \\
& x_t^d \leq \theta_{max}.
\end{aligned}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{z}_t$ are the system states and observations at time t respectively, K is the number of cameras, $\boldsymbol{z}_t^{(k)}$ is the image observation for the k^{th} camera at time t and $L \subset \{LH_{silhouette}, LH_{edge}\}$ is a set of likelihood functions, x_t^d is the d^{th} element of the state vector $\boldsymbol{x}_t$ representing the d^{th} joint angle, D is the dimensionality of the problem, θ_{min} and θ_{max} are the anatomic joint angle limits of the d^{th} joint angle.

4.1.2 Solution Methods

The algorithms available for solving COPs can be divided into exact and approximate ones. Analytical approaches yield exact solutions. They can also find all minima and maxima, including the global minimum and maximum, if they exist. However, these approaches can only be used where a mathematical formulation of the cost function is available. Unfortunately, in most real world problems only sample empirical data is available.

Other methods for continuous optimization can only obtain approximate solutions. Derivative-based methods find local minimum reasonably quickly. They use numerical differentiation which only requires the value of the function at any point in the search space rather than the symbolic formula of the function. But these methods also require evaluation of the derivatives of the cost function. First derivative or the first and the second derivatives are required depending upon the method employed.

Another type of approximate methods are the direct search methods. They are more robust than the above two type of methods, and therefore able to effectively deal with a wider range of COPs. They only require the function to be evaluated at points within the search space. However, like all approximate methods, they only find the local optimum.

Finally, a recent trend is the use of metaheuristics for the solution of COPs. Metaheuristics do not provide any guarantee about the quality of found solution. However, they may be able to look both for local as well as global minima. They also have very little requirement about the problem formulation. Similar to the direct search methods, they only need to be able to evaluate the function at any point in the search space. In particular, population-based optimization algorithms have attracted much attention for a variety of optimization problems and have shown better performance then derivative-based optimization methods [7, 6, 10, 11, 8]. They do not require the objective function to be differentiable or even continuous. They generally perform better as global optimization techniques due to their balance between exploration and exploitation of the whole solution space. An AP-PSO algorithm is proposed and implemented in this chapter to demonstrate the effectiveness of population-based optimization in AHT problem domain.

4.2 Particle Swarm Optimization

PSO is a new population based stochastic optimization technique which has shown considerable success in solving non-linear, non-differentiable, multimodal optimization problems [65]. PSO updates the particle position towards the optimal point by updating the position and velocity of each particle according to the best value of each particle and the global best value of all the particles. The particle with the best fitness value denotes the optimal point in the search space. The computation of PSO is easy and adds only a slight computational load when it is incorporated into the generic PF framework. It also has the flexibility to control the balance between local and global exploration of the problem space which helps to enhance the search ability and derives the particles towards high likelihood regions, thereby helping to overcome the degeneracy issue in generic PF.

4.2.1 Basic Algorithm

PSO method is inspired by swarm behavior such as a bird flocking [65]. Each particle updates its position and direction of motion on the basis of its current direction of motion, its best position in the past and the best position of the entire swarm. The particles move by searching for the best position that gives the highest fitness (cost function evaluation) value.

Let us consider a d dimensional search space with N-particles. The position of the i^{th} particle at the k^{th} generation (iteration) is denoted by $x_k(i) \in \Re^d$. Its velocity and fitness value is represented by $v_k(i) \in \Re^d$ and $J_k(i) \in \Re$ respectively. The particle's best previous position and fitness is denoted by $p_k(i) \in \Re^d$ and $J_{pk}(i) \in \Re$ respectively. The global best position and fitness value of the whole swarm is denoted by $g_k \in \Re^d$ and $J_{gk} \in \Re$.

Each particle updates its velocity and position according to the following equations:

$$v_{k+1}(i) = w \cdot v_k(i) + c_1 \cdot rand(\,) \cdot (g_k - x_k(i)) + c_2 \cdot rand(\,) \cdot (p_k(i) - x_k(i)), \quad \forall i \tag{4.1}$$

$$x_{k+1}(i) = x_k(i) + v_{k+1}(i) \tag{4.2}$$

where w is an inertia weight, $rand()$ is a uniformly distributed random number in the range $[0, 1]$, c_1 and c_2 are positive learning factors. These are termed as social (global) and cognitive (local) learning factors respectively. The particle position update procedure is illustrated in Figure 4.1.

The parameters play an important role in directing the exploratory behavior of the swarm. Higher values of the inertia weight will force the particles to explore a wider region of the search space and lower inertia weights and learning factors will help to focus search in a smaller area. Assigning excessively large values to the inertia weight will cause a divergent behavior of the swarm and the performance will not be better than a random search [69]. The typical values for these parameters are $w = 0.7, c_1 = c_2 = 1.5$ [65]. The schematic of the PSO algorithm is displayed in Figure 4.2.

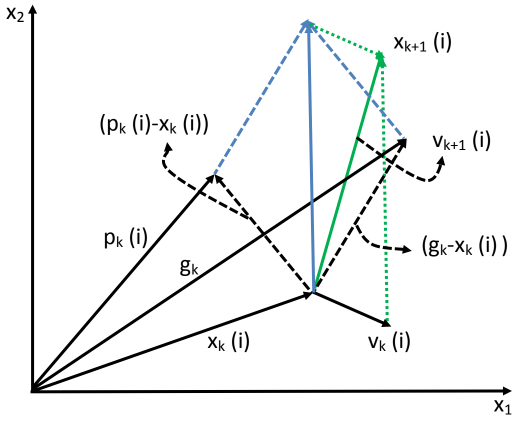


Figure 4.1: Particle position update using PSO

The PSO is implemented in PF framework to alleviate some of the PFs inherent problems. The Sequential Importance Sampling (SIS) (see Appendix B, Section B.2) method of particle filtering draws samples from the importance/proposal distribution. The most common choice of the proposal distribution is the probabilistic/process model of the state evolution which will result in an approximate posterior distribution. Since the proposal function of generic PF is suboptimal, it will pose two major challenges: degeneracy, or particle impoverishment, and sample size dependency. Degeneracy occurs when likelihoods are very narrow. It also occurs when the observation lies in the tail of the prior distribution as shown in Figure 4.3. In this case, all but a few particles will have insignificant weights. Only the particles with significant weights will be selected for resampling. As a result, there will be only few dissimilar particles which will result in the loss of diversity of solution space search.

If the sample size is small, which will improve the computational efficiency, then there is a danger that the particles might not be distributed around the true states. As a result after a few iterations, it will become difficult for the particles to converge to the true states.

4.3 Recent Contributions to Alleviate Degeneracy

To solve this issue, Rudolph et al. [121] proposed the Unscented Particle Filter (UPF) technique by combining the UKF with a generic PF to get a better proposal function. It results in improved performance but sacrifices computational efficiency. Another method

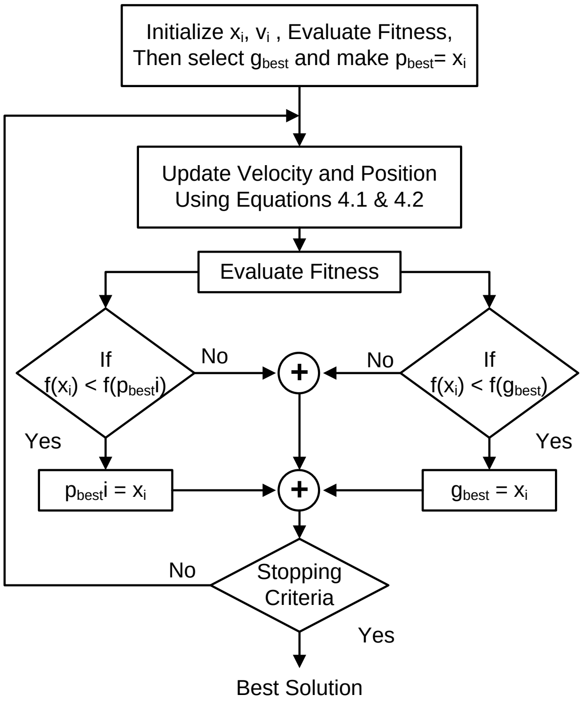
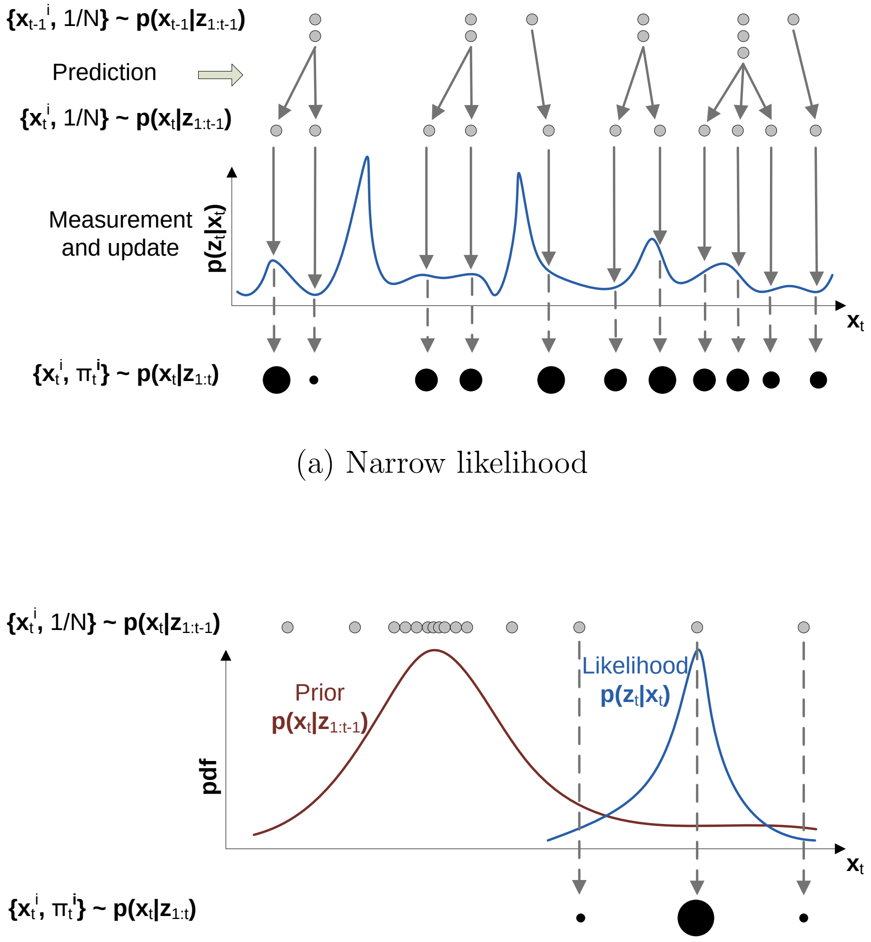

Initialize x_i, v_i , Evaluate Fitness, Then select g_{best} and make p_{best}= x_i
Update Velocity and Position Using Equations 4.1 & 4.2
Evaluate Fitness
If $f(x_i) < f(p_{best}i)$
No
No
If $f(x_i) < f(g_{best})$
Yes
Yes
$p_{best}i = x_i$
$g_{best} = x_i$
No
Stopping Criteria
Yes
Best Solution

Figure 4.2: Schematic of PSO algorithm

is to use an intelligent algorithm to optimize the sampling process. Clapp et al. [21] introduces simulated annealing into PF. The method uses an annealing function based on the observations, which results in improved performance when the likelihood lies in the tail of the prior distribution. Higuchi et al. [143] combines Genetic Algorithm (GA) and sequential MC to introduce diversity in the population after resampling. Torma et al. [152] added local search methods into PF to reduce the sample size and improve efficiency when observations are reliable.

Variants of PSO have also been developed specifically to address AHT. Wang et al. [163, 165] proposed an Annealed PSO based PF algorithm (APSOPF) for whole body human tracking. In this algorithm, the sampling covariance and annealing factors are incorporated into the velocity update equation of PSO. The parameters/coefficients are initialized with appropriate values in the beginning of PSO iteration and are decreased (annealed) in small steps. They reported that the use of the sampling covariance has constrained the particles to most likely regions of pose space and annealing has preserved particles own divergence and self-exploration capabilities before convergence.


$\{x_{t-1}^i, 1/N\} \sim p(x_{t-1}|z_{1:t-1})$
Prediction
$\{x_t^i, 1/N\} \sim p(x_t|z_{1:t-1})$
Measurement and update
$p(z_t|x_t)$
x_t
$\{x_t^i, \pi_t^i\} \sim p(x_t|z_{1:t})$
$\{x_t^i, 1/N\} \sim p(x_t|z_{1:t-1})$
Prior
$p(x_t|z_{1:t-1})$
Likelihood
$p(z_t|x_t)$
pdf
x_t
$\{x_t^i, \pi_t^i\} \sim p(x_t|z_{1:t})$

(a) Narrow likelihood

(b) Likelihood lies in the tail of the prior distribution

Figure 4.3: Degeneracy scenarios of generic PF

Kwolek et al. [71] proposed a global-local annealed PSO to minimize the mismatch between the observed human pose and the estimated pose. At the beginning of the optimization cycle a pool of candidate solutions using APF is generated and among this pool the global best solution is selected. Later on, each limb pose is optimized locally with a smaller number of particles to refine this global best solution. They reported that their algorithm outperforms the original PSO algorithm.

Zang et al. [172] proposed a hybrid search method, Niching Swarm Filter (NSF), which involves a refinement step of local optimization. A non-parameter niching method ring topology based on bare bones PSO algorithm is integrated in the PF framework. The niching search process can help to robustly and efficiently find multiple significant modes (both local and global peaks) of the configuration distribution. A local search further refines the proposed pose estimate.

John et al. [60, 61] proposed a Hierarchical PSO (HPSO) for full body AHT from multi-view video sequences. They employed a 12-step hierarchical method for locating the body parts. The method starts with the estimate of global position and orientation and step-by-step estimates torso, left upper-arm, left lower-arm, right upper-arm, right lower-arm, head, left upper-leg, left lower-leg, right upper-leg and right lower-leg. They reported their algorithm outperforms PF and APF.

Nguyen et al. [98] proposed hierarchical annealed PSO algorithm. They claimed improved search efficiency due to hierarchical search and alleviation of noisy observation problem by incorporating annealing factor term into the velocity update equation of PSO. They claimed efficiency and effectiveness of their approach both in terms of tracking accuracy and computation time.

4.3.1 Parameter Adaptation of PSO

A plethora of research literature has investigated the parameter adaptation of PSO algorithm. Most of these are related to the tuning of inertia weight parameter. The first set of these methods uses a time varying inertia weight strategy [3, 76, 58]. The second set of methods uses feedback to monitor the state of the algorithm and adjust inertia weight on these basis [56, 166].

Some adaptive parameter PSO schemes are also proposed to update the inertia weight as well as cognitive and social learning factors. These schemes use different strategies for parameter adaptation. Tang et al. [145] designed a quadratic function for inertia weight update and decreased learning factors on the basis of generation number and particles' best fitness and later used a stochastic learning method to refine the solution. Ratnaweera et al. [115] used time varying inertia weight and learning factors. They also introduced mutation to the selected modulus of the velocity vector. In another strategy they used social, like Bat algorithm velocity update (Chaper# 5), or cognitive component only for velocity update. Tripathi et al. [151] used time varying update of inertia weight and learning factors and applied it for multi-objective optimization. Juang et al. [62] used fuzzy set theory to adaptively update learning factors and incorporated it with quadratic interpolation and crossover operator to enhance the global search capability of PSO.

Ismail et al. [55] used a secondary swarm for parameter optimization. Hashemi [51] used three learning automata, one for each parameter, to determine the values of the constants in the velocity update equation. Montalvo et al. [92] added w, c_1 and c_2 as 3-additional dimensions to the problem and estimate them using PSO and later used them. Niu et al. [99] linearly decreased the constants of the velocity update equation with iterations. Wang [161] decreased w and c_1 between a maximum and minimum value with iterations. For c_2, he used the same scheme but with a mutilative factor depending on best and worst fitness values. Alireza [9] used a mutation constant depending on the best fitness to control the rate of mutation of the selected dimension. Furthermore, w is varied between 0.5 and 1 using best fitness of the solution.

4.4 Proposed Adaptive Parameter PSO Algorithm

A new PSO algorithm, AP-PSO algorithm, is proposed in this work which adaptively updates its inertia weight and social and cognitive learning factors. The inertia weight depends on maximum variation allowed and varies with the number of PSO iterations, social and cognitive learning factors vary according to the fitness value.

Initial inertia weight value is set to half of the maximum variation allowed in a single time step in each problem dimension and then decreased by an annealing factor which is dependent on the PSO iterations. The inertia weight parameter is varied according to:

$$w = \frac{V_{max}}{2} e^{(1-\frac{m}{M})} \tag{4.3}$$

where V_{max} is the maximum variance allowed in each time step in each dimension, M is the maximum allowed number of PSO iterations and m is the current iteration. It is clear that as the current iteration value increases, the inertia weight value decreases and helps the swarm to settle around the best solution.

The proposed algorithm also allows the learning factors c_1 and c_2, which influence the social and cognition components of swarm behavior, to vary according to the quality of the obtained solution. Social learning factor is varied according to:

$$c_1 = \lceil 0.7e^{(1-J_k(i))} \rceil \tag{4.4}$$

which indicates that the contribution of the social factor in velocity update will be large when the particle is far away from the global best solution. The influence of the social component will be reduced to allow the search to concentrate on a more local region, when the particle is in the vicinity of the global best position.

The cognition learning factor is varied according to:

$$c_2 = \lceil e^{(J_{pk}(i)-J_k(i))} \rceil \tag{4.5}$$

Similarly, if the particle's position has improved with respect to its own best position, its an indication that the particle is moving in the right direction and the change in velocity due to this component should be lowered. Whereas, a deterioration in particle evaluation indicates that it is moving in the wrong direction and hence requires a change in velocity. The ceiling for both social and cognition learning factors is set at $c_1 = 1.5$ and $c_2 = 1.5$. A comparison of the proposed adaptive parameters to basic PSO parameters is given in Table 4.1.

Table 4.1: Comparison of fixed and adaptive PSO parameters

Algorithm	w	c_1	c_2
PSO	0.749	1.5	1.5
AP-PSO	$\frac{V_{max}}{2}e^{(1-\frac{m}{M})}$	$\lceil 0.7e^{(1-J_k(i))} \rceil$	$\lceil e^{(J_{pk}(i)-J_k(i))} \rceil$

The schematic of the proposed AP-PSO algorithm is displayed in Figure 4.4 and its pseudocode is given in Algorithm 1.

4.5 Experimental Results

To evaluate the performance of the proposed AP-PSO algorithm, experiments were conducted to identify the parameters of two simple oscillators and a 10 dimensional human arm tracking problem. These results are presented in the following sections.

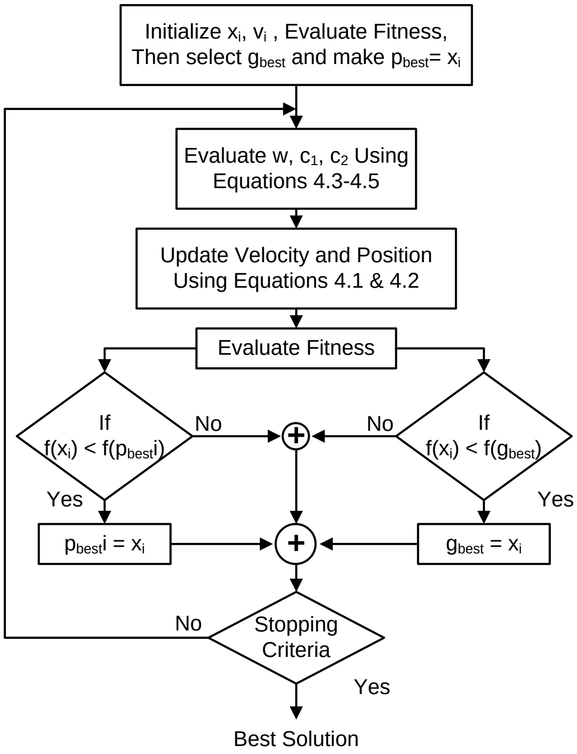

Initialize x_i, v_i , Evaluate Fitness, Then select g_{best} and make p_{best}= x_i
Evaluate w, c_1, c_2 Using Equations 4.3-4.5
Update Velocity and Position Using Equations 4.1 & 4.2
Evaluate Fitness
If $f(x_i) < f(p_{best}i)$
No
No
If $f(x_i) < f(g_{best})$
Yes
Yes
$p_{best}i = x_i$
$g_{best} = x_i$
No
Stopping Criteria
Yes
Best Solution

Figure 4.4: Schematic of the proposed AP-PSO algorithm

4.5.1 System Identification

In the first set of experiments, a parameter identification task was performed. Two second order linear time invariant oscillators were considered. The first system is described as

$$G_1(s) = \frac{Ke^{-T_d s}}{T_2 s^2 + T_1 s + 1} = \frac{7.6e^{-25s}}{10s^2 + s + 1} \tag{4.6}$$

The goal was to identify the DC gain K, delay time T_d and the natural period of oscillation T_2 from the training data.

The second system was described as

$$G_1(s) = \frac{5.7e^{-42s}}{40.2s^2 + 4.2s + 1} \tag{4.7}$$

For this system, the DC gain K, delay time T_d, natural period of oscillation T_2 as well as the damping parameter T_1 were identified.

Algorithm 1 Adaptive parameter particle swarm optimization algorithm

```
Randomly initialize $x_i$ and $v_i$
Evaluate fitness $f(x_i)$
Set $p_{best} = x_i$ and select $g_{best}$
repeat
    for all particles $i$ do
        Update parameters $w$, $c_1$ and $c_2$ using Equations 4.3-4.5
        Update Velocities and positions using Equations 4.1-4.2
        Evaluate fitness of updated particles
        if $f(x_i) < f(p_{best}i)$ then $p_{best}i = x_i$;
        if $f(x_i) < f(g_{best})$ then $g_{best} = x_i$;
        Update $x_i$ and $v_i$
    End for
until Stopping criteria
```

The outputs obtained by numerically integrating the system equations were used as the training data set. PF, PSO and AP-PSO algorithms were used to identify the parameters. The parameters identified by the optimization algorithms were used to reproduce the system models. Their output was then compared with the training data. The absolute difference between the true and identified models at each time step were added together to form an integral absolute error that was to be minimized during the course of optimization.

Real life measurements are rarely free of noise. To emulate this behavior, a zero mean Gaussian noise was added to the noise free training data. The Signal to Noise Ratio (SNR) was $\sim 24dB$. That is the power of noise was approximately one eighth of the signal power. The three algorithms were also used to identify the parameters from this noisy training data. Figure 4.5 illustrates the types of training data used in these experiments.

The fitness evaluations were fixed to 250 for all three algorithms to achieve a fair evaluation of algorithm performance. PF was assigned 250 particles, PSO and AP-PSO were assigned 25 particles and 10 iterations of PSO runs were used. Initial particle velocities were randomly set to small values using a zero mean Gaussian model. The initial populations of particles were generated using a Gaussian model by setting the means and standard deviations to half of the parameter values coarsely guessed from the clean training data.

Since the optimization algorithms are highly stochastic, the experiments were run 20 times for each set of training data. The mean, standard deviation and parameter values corresponding the best normalized fitness values are listed in Tables 4.2 and 4.3. A smaller

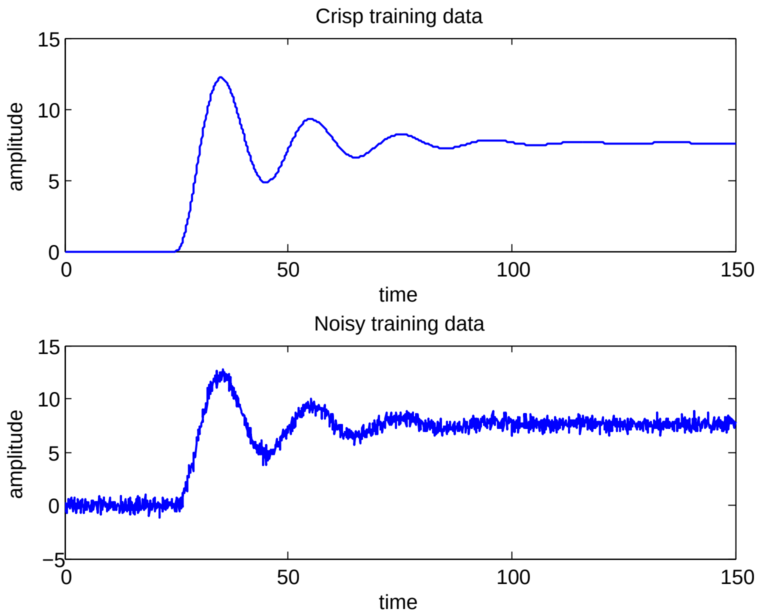
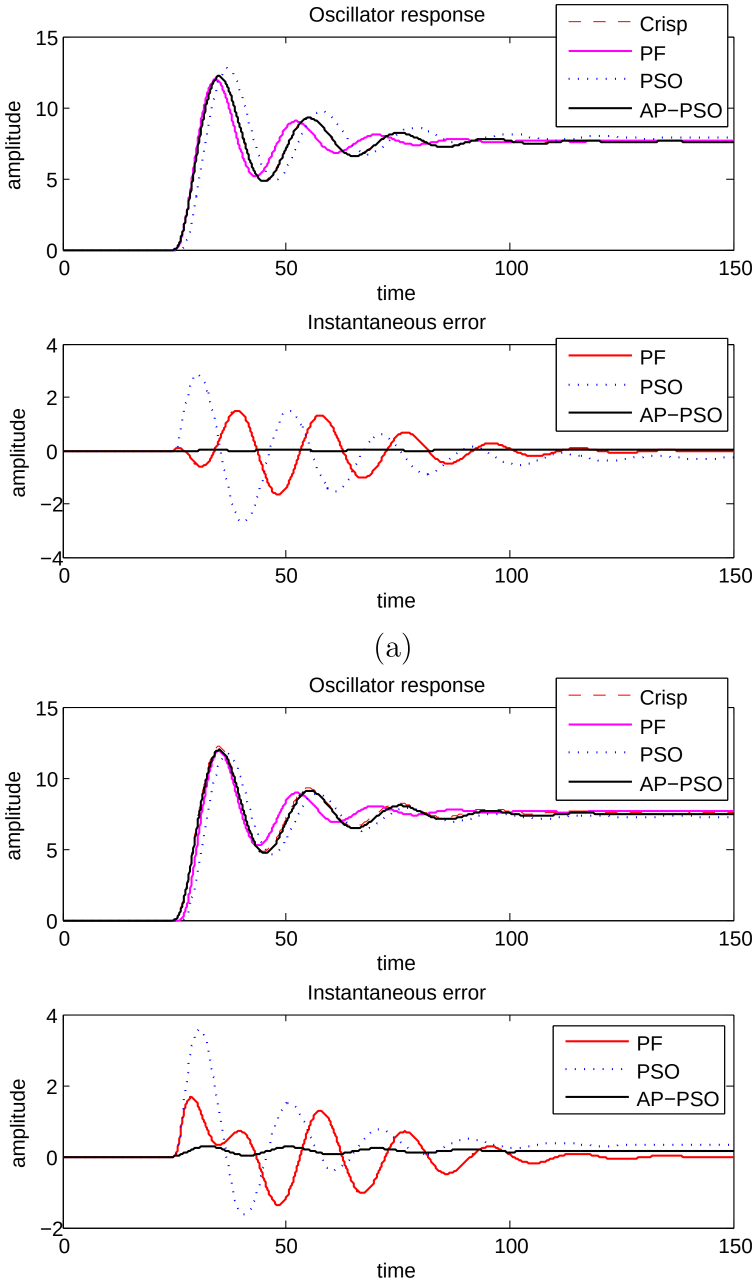


Figure 4.5: Typical samples of input data used in training

fitness value represents a better solution as the integral absolute error is minimized over the simulation time.

Response of the two systems obtained from the parameters identified by three algorithms during an experimental run is illustrated in Figures 4.6 and 4.7. The plots in Figures 4.6(a) and 4.7(a) represent the responses when parameters were identified using clean training data and the plots in Figures 4.6(b) and 4.7(b) depict the responses when the parameters were identified from noisy training data. The corresponding errors between the clean training data and the obtained system responses is also plotted in the corresponding figures.

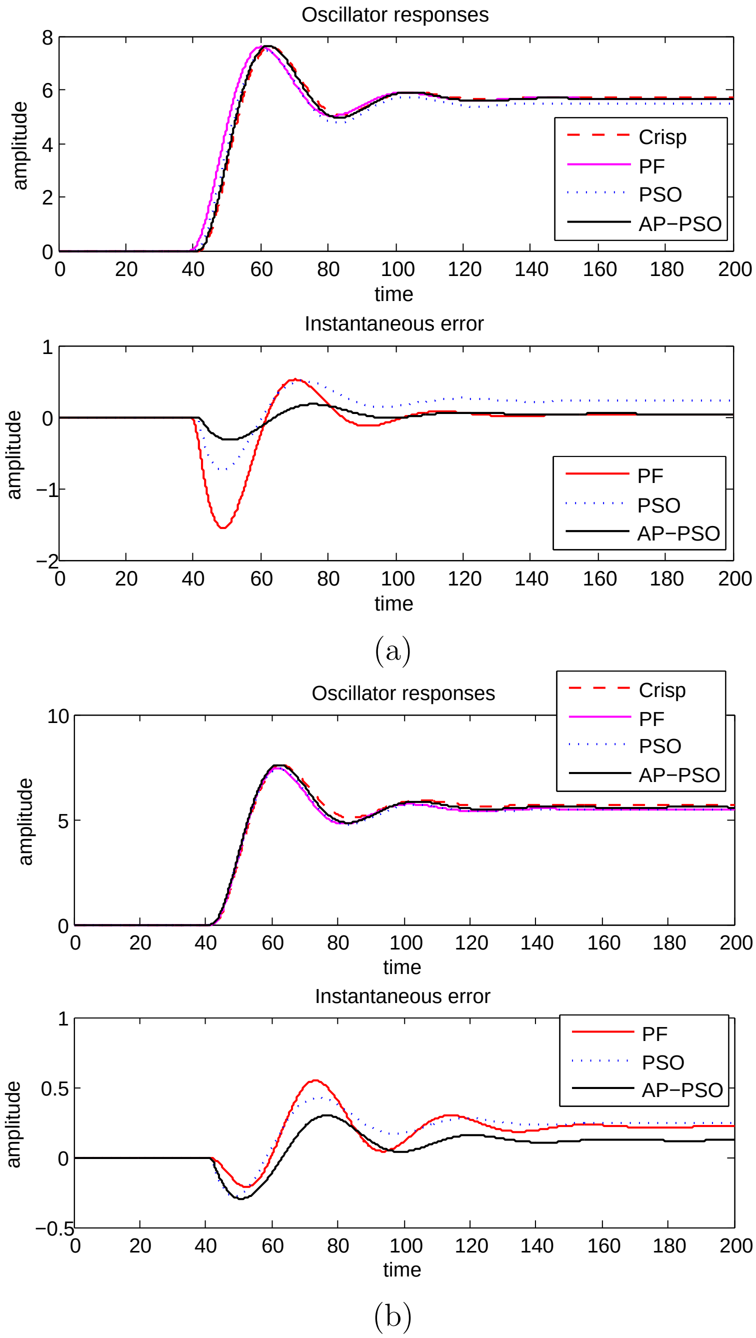
From the obtained results, it is clear that the AP-PSO has performed the best among the three considered algorithms in terms of mean parameter identification value. The performance is particularly very good for parameter identification of first system. This is due to the fact that all three parameters to be identified has a significant effect on the error measure that was being minimized. Any deviation from the true parameter values has resulted in an increase in error and hence the activation of the search. For the second system, the first three parameters are adequately identified by AP-PSO with good repeatability. However, the fourth parameter T_1 has more percentage variation ($\sim 20\%$). This parameter is related to the damping ratio of the system which is mainly shaping the



(a)

(b)

Figure 4.6: Response of the first oscillator to a step input utilizing the parameters identified by PF, PSO and AP-PSO (a) Parameters identified using crisp training data (b) Parameters identified using noisy training data



(a)

(b)

Figure 4.7: Response of the second oscillator to a step input utilizing the parameters identified by PF, PSO and AP-PSO (a) Parameters identified using crisp training data (b) Parameters identified using noisy training data

Table 4.2: Comparison of PF, PSO and AP-PSO results for $G_1(s)$ parameter identification

Data	Algo.	Parameter	True	Mean	STD	Best	$f(Best)$
Crisp	PF	K	7.6	7.6233	0.0800	7.5539	0.052
		T_d	25	26.2343	0.6593	25.8890	
		T_2	10	7.6862	0.2325	7.9884	
	PSO	K	7.6	7.5798	0.1967	7.5645	0.0029
		T_d	25	25.2431	0.7695	25.6367	
		T_2	10	9.6269	0.7677	9.3289	
	AP-PSO	K	7.6	7.6061	0.1137	7.5992	0.0003
		T_d	25	25.3054	0.5361	25.0106	
		T_2	10	9.7093	0.6701	9.9906	
Noisy	PF	K	7.6	7.6409	0.0773	7.5929	0.063
		T_d	25	26.3480	0.6288	25.7357	
		T_2	10	7.6611	0.1654	7.4405	
	PSO	K	7.6	7.5101	0.2068	7.5946	0.0095
		T_d	25	25.4396	0.8758	26.2874	
		T_2	10	9.9637	0.7044	9.0066	
	AP-PSO	K	7.6	7.5567	0.2333	7.6486	0.0073
		T_d	25	25.0362	0.4376	25.1437	
		T_2	10	10.0037	0.5904	9.6485	

transient response and is affecting the error measure only for a short period of time, $t \sim 50$ to $t \sim 100$ seconds. Change in absolute integral error due to little change in T_1 is less pronounced than varying any other parameter. Moreover, PSO algorithm inherently has a poor local search ability due to the random velocity effect. As a result error does not seem to decrease after a certain value.

In order to improve the identification of damping parameter, a harmonic input was applied to the second oscillator and the corresponding output was used for training. The identified parameters by PF, PSO and AP-PSO using this harmonic input are listed in Table 4.4 and the system responses with corresponding errors are shown in Figure 4.8. Application of harmonic input has resulted in a better estimation of the damping parameter but the estimation error of the DC gain parameter K was increased.

In the last parameter identification experiment, a mixture of step and harmonic inputs was applied to the second oscillator and the corresponding output was used for training. This combination has resulted in an optimal identification of system parameters that captures both transient and steady state system behavior. The results are listed in Table 4.5

Table 4.3: Comparison of PF, PSO and AP-PSO results for $G_2(s)$ parameter identification

Data	Algo.	Parameter	True	Mean	STD	Best	$f(Best)$
Crisp	PF	K	5.7	5.8590	0.0515	5.8172	0.018
		T_d	42	41.1953	1.0025	40.9326	
		T_2	40.2	37.8083	1.5769	37.8889	
		T_1	4.2	4.5099	0.5372	4.0412	
	PSO	K	5.7	5.3221	1.5794	5.7601	0.003
		T_d	42	41.1812	1.9747	41.8509	
		T_2	40.2	38.9753	1.7848	40.2239	
		T_1	4.2	4.5227	1.3932	3.7684	
	AP-PSO	K	5.7	5.7050	0.0894	5.6989	0.0011
		T_d	42	41.9946	1.1262	42.7816	
		T_2	40.2	38.4976	1.9090	37.638	
		T_1	4.2	4.1697	0.9805	4.1495	
Noisy	PF	K	5.7	5.8998	0.1067	5.7935	0.015
		T_d	42	40.6750	1.1137	41.5389	
		T_2	40.2	37.6144	1.2010	36.3683	
		T_1	4.2	4.6179	0.5806	4.0076	
	PSO	K	5.7	5.6584	0.2234	5.7308	0.0044
		T_d	42	41.8481	1.3304	42.6995	
		T_2	40.2	38.4329	2.0414	37.8194	
		T_1	4.2	4.3877	0.7389	4.4383	
	AP-PSO	K	5.7	5.6767	0.1007	5.6545	0.0028
		T_d	42	41.8952	1.2901	42.8998	
		T_2	40.2	38.2418	2.0621	37.2930	
		T_1	4.2	4.3722	0.8646	4.4732	

Table 4.4: Identified parameters using harmonic input training data

Parameter	True	PF	PSO	AP-PSO
K	5.7000	3.7554	5.8835	5.8299
T_d	42.0000	42.0092	41.7713	42.0033
T_2	40.2000	34.0202	41.5025	41.0012
T_1	4.2000	3.7988	4.5042	4.4129

and the system responses and errors are depicted in Figure 4.9.

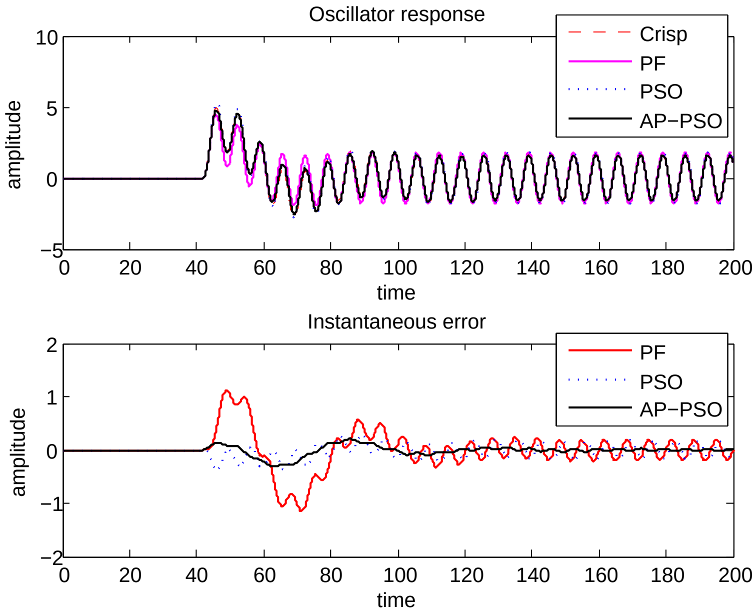


Figure 4.8: Oscillator response and corresponding errors when a harmonic training data is used

Table 4.5: Identified parameters using a combination of step and harmonic input training data

Parameter	True	PF	PSO	AP-PSO
K	5.7000	5.1080	5.7090	5.7060
T_d	42.0000	42.2005	42.1710	41.9845
T_2	40.2000	37.3877	40.9584	41.5851
T_1	4.2000	4.8643	4.4986	4.2671

4.5.2 Qualitative Analysis: Human Arm Tracking

In this experiment, a qualitative analysis of human arm tracking based on visual inspection is performed. The human arm and torso are modeled as a set of cylinders connected by revolute joints [129],[12]. The body parts included are torso, upper and lower arms. Shoulder joint has 3-degrees of freedom and the elbow joint has 1-degree of freedom. There are 6-degrees of freedom for the global position and orientation. Overall, this results in a 10-dimensional state-space representation of the human arm shown in Figure 4.10. The motion of a point in the body part is determined by forward kinematics.

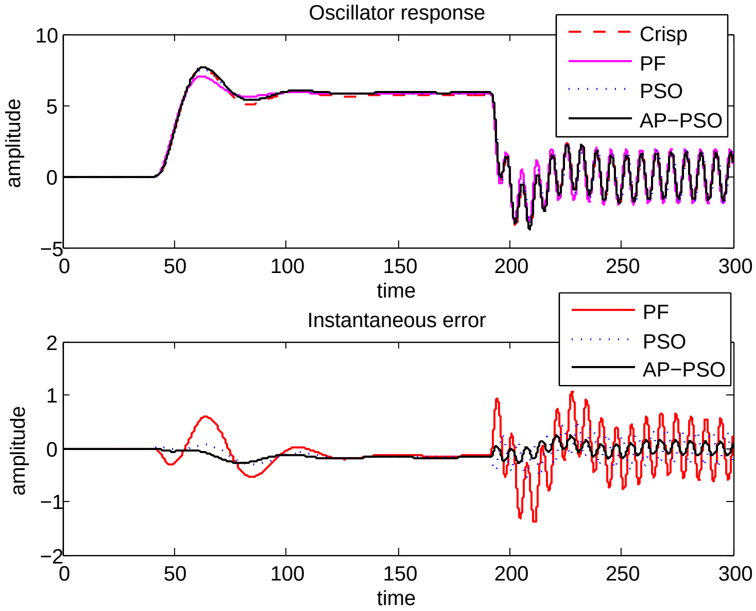


Figure 4.9: Oscillator response and corresponding errors when a combination of step and harmonic training data is used

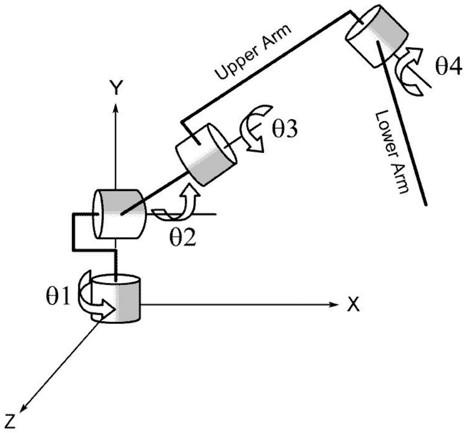


Figure 4.10: The human arm model [129]

The experiment considers the video of a person moving his right arm in front of a camera. The video is recorded at 25 frames per second and a total of 80 frames are used. The frame size is 240x320 pixels. Background subtraction is used to extract the object of interest. Camera calibration data were experimentally obtained for the rotation and translation transformation vectors. These vectors are used to map data from the world coordinate system to the camera reference frame [153].

The experiments test the use of a generic PF and the proposed AP-PSO for human arm tracking. A constant velocity temporal model is employed to predict the joint angles, states of the model. For AP-PSO experiments, 10-iterations of PSO runs are used. It is found that, the increase in number of particles has resulted in improved tracking performance for both algorithms.

A comparison of the results of generic PF and AP-PSO are shown in Figure 4.11. The first column shows PF results when 2000 particles are used and the second column shows the AP-PSO results when 200 particles are used. To maintain the same number of likelihood evaluations across the two algorithms, we set the number of particles in PF to 10 times that in AP-PSO. Similarly, the third and fourth columns show the PF and AP-PSO results with 5000 and 500 particles respectively. Visual inspection shows that the AP-PSO outperforms the generic PF in tracking the human arm. This can be deduced by comparing the third and fourth rows of Figure 4.11 that over time PF tracking of the lower arm and elbow diverges in Frame# 74 and 78, while AP-PSO continues to track them.

Figure 4.12 shows the results of AP-PSO when different numbers of particles are employed for tracking. Visual inspection indicates that with the increase in the number of particles the estimation accuracy is improved at the expense of added computational cost.

It is observed from the results that when a video sequence captured by a single camera, monocular video, is used for articulated tracking, due to the loss of depth information and self-occlusions, the orientation results are not accurate in the absence of an adequate occlusion model. In the quantitative analysis, video sequences recorded by multiple cameras are used to alleviate loss of depth and self-occlusion challenges.

4.6 Conclusions

In this chapter, an improved PSO algorithm, AP-PSO, is proposed that uses the quality of obtained results to update the inertia weight, cognitive learning factor and social learning factor. This improvement has imparted good exploration properties to the proposed AP-PSO algorithm. It favors a local search of the solution space as the particles comes into

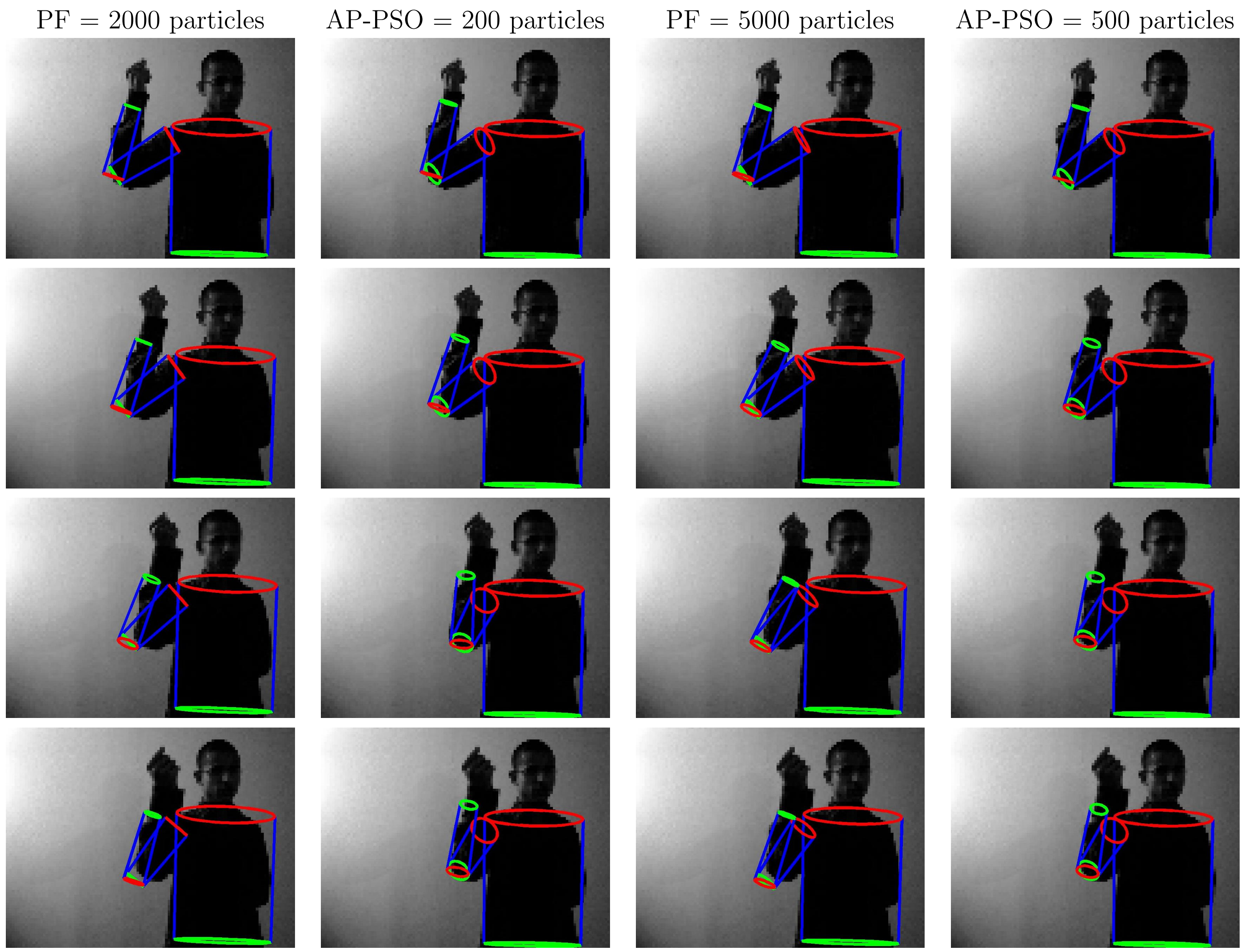


Figure 4.11: Comparison of PF and AP-PSO results Frame#66,70,74,78

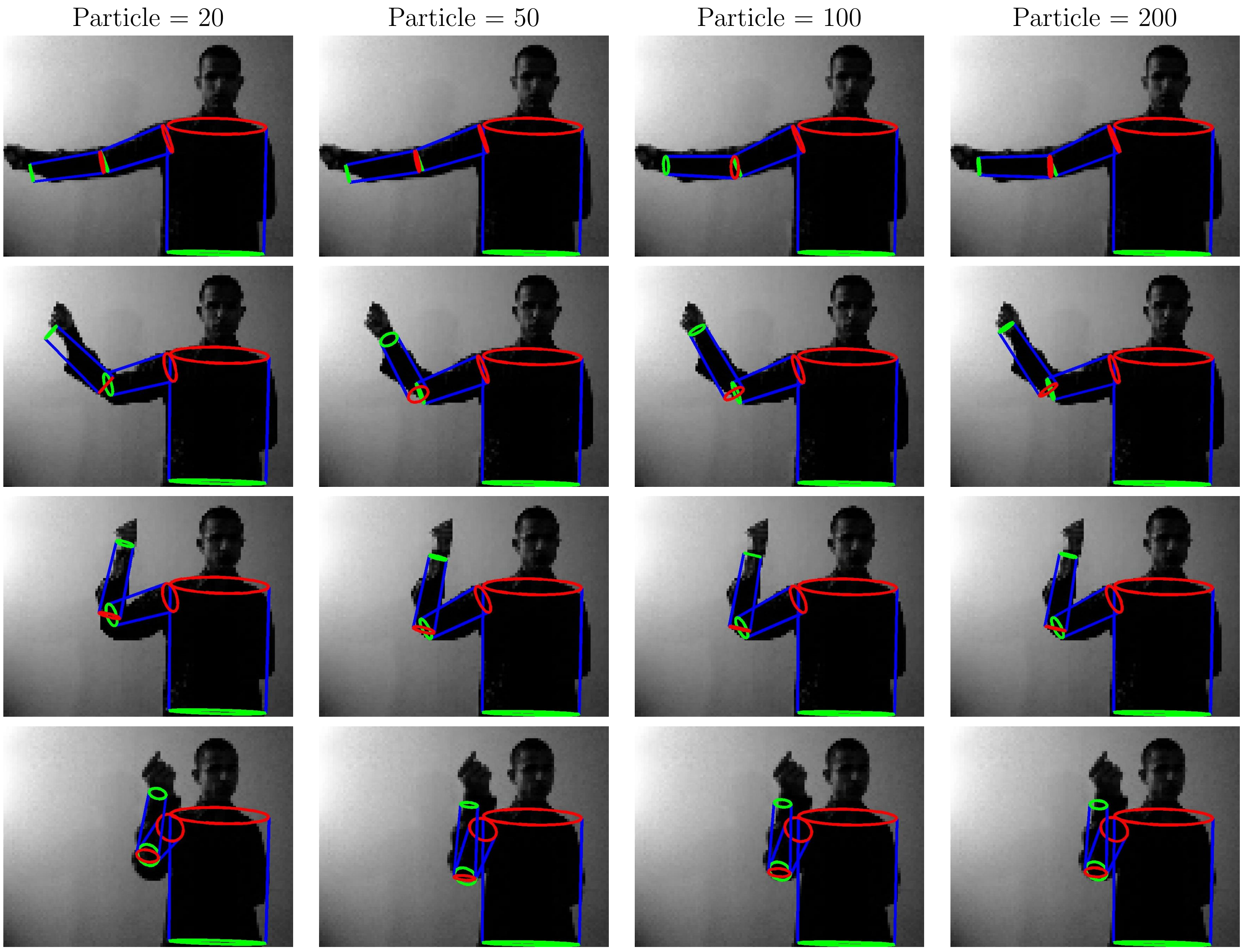


Figure 4.12: Results of AP-PSO tracking with different populations Frame#20,40,60,80

the vicinity of the optimal solution and favors a global search if the particle is away from the optimal solution. Experiments are run to evaluate the performance of this proposed algorithm. The preliminary quantitative and qualitative analysis results depict improved performance of the proposed algorithm compared to PF, APF and PSO algorithms.

Chapter 5

Modified Artificial Bat Algorithm

In this chapter, the echolocation feature of microbats is used to develop a novel metaheuristic optimization algorithm. Section 5.1 describes how microbats use echolocation to locate their prey and a method to incorporate this property to solve optimization problems. The basic Artificial Bat (AB) algorithm is also explained. Section 5.2 briefly describes some of the latest developments and applications of AB algorithm. Section 5.3 describes the proposed Modified Artificial Bat (MAB) algorithm. Section 5.4 explicates the experimental results of applying the developed optimization algorithm on benchmark function evaluations. Finally, Section 5.5 concludes the chapter with discussion of the obtained results.

5.1 Echolocation

5.1.1 Behavior of Natural Bats

The proposed algorithm is inspired by echolocation abilities of bats. There are many different species of bats and they are of different sizes. Among them, microbats extensively use echolocation [117, 116]. They use a type of sonar to detect prey, avoid obstacles, and locate their roosting crevices in the dark. They emit loud sound pulses and listen for the echo that bounces off their surrounding objects. These sound pulses vary in properties and different species emit pulses of different bandwidth.

They emit sound pulses of constant frequency in the range of 25kHz to 150kHz. Each ultrasonic sound burst lasts for a very short period of time, typically 5 to 20 ms. Microbats normally emit about 10-20 sound bursts every second. The rate of emission of these sound pulses increases, to about 200 pulses per second, when they fly close to their prey. While searching for prey, bats produce loud sound pulses in the range of 110dB but when bats get closer to their prey, they become quieter. Such echolocation capabilities of microbats can be associated with the objective function to be optimized and optimization algorithms can be formulated that mimic this bat behavior in finding optimal solutions.

5.1.2 Basic Bat Algorithm

Bat-inspired algorithms, bat algorithms, can be developed by idealizing some of the echolocation characteristics of microbats. The following assumptions are made to approximate bat's echolocation properties to solve an optimization problem [168]:

1. All bats use echolocation to sense distance.
2. Bats fly randomly with velocities $\boldsymbol{v}_i$ at position $\boldsymbol{x}_i$ with a fixed frequency f_{min} by varying wavelength λ and loudness A_0 to search for prey.
3. Depending on the proximity of the prey, they adjust their frequency and can adjust the pulse emission rate $r_i \in [0 - 1]$.
4. Also depending on the proximity of the prey, their loudness vary from a large A_0 to a small A_{min} values.

In practical implementations, frequency is limited to a range $[f_{min}, f_{max}]$ and is chosen such that it is comparable to the size of the domain of interest.

Movement of Bats: Generation of New Solutions

For a virtual bat to solve an optimization problem, rules need to be defined to set their positions and velocities in the d-dimensional search space. The new position $\boldsymbol{x}_i^t$ and velocity $\boldsymbol{v}_i^t$ at time step t are given as:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{5.1}$$

$$\boldsymbol{v}_i^t = \boldsymbol{v}_i^{t-1} + (\boldsymbol{x}_i^{t-1} - \boldsymbol{x}^*)f_i \tag{5.2}$$

$$\boldsymbol{x}_i^t = \boldsymbol{x}_i^{t-1} + \boldsymbol{v}_i^t \tag{5.3}$$

Here $\beta \in [0-1]$ is a random vector drawn from uniform distribution, $\boldsymbol{x}^*$ is the current global best solution among the bat population. Accordingly, following Equation 5.1 each bat is assigned a frequency in the range $[f_{min}, f_{max}]$ that determines the bat's velocity.

Local Search

A bat is randomly selected for local search. The bat selected for local search generates a new solution on the basis of its current loudness A_i and current position as:

$$\hat{\boldsymbol{x}}_i^t = \boldsymbol{x}_i^t + \varepsilon A_i \tag{5.4}$$

where $\varepsilon \in [-1, 1]$ is a random number.

Loudness and Pulse Emission

The loudness usually decreases once a bat has found its prey, while rate of pulse emission increases. The loudness and pulse emission rate are only varied once a solution is improved. The bat moves towards optimal solution according to:

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0[1 - e^{-\gamma t}] \tag{5.5}$$

where α and γ are constants. In fact, α is like the cooling factor in a simulated annealing scheme. For $0 < \alpha < 1$ and $\gamma > 0$, we get

$$A_i^t \to 0, \quad r_i^t \to r_i^0, \quad as \quad t \to \infty$$

Loudness is initially set to the range $A_i \in [0.1 - 0.9]$ and emission rate is set to the range $r_i \in [0-1]$ and the constants are set to $\alpha = \gamma = 0.9$. A_i decreases as the solution improves resulting in a more precise local search as evident by Equation 5.4.

The velocity and position update of AB algorithm has similarities with standard PSO. In AB algorithm f_i controls the speed and the range of swarm movement. However, AB algorithm provides local search controlled by loudness and pulse emission rate. Based on the above idealization, a flow chart of AB algorithm is given in Figure 5.1.

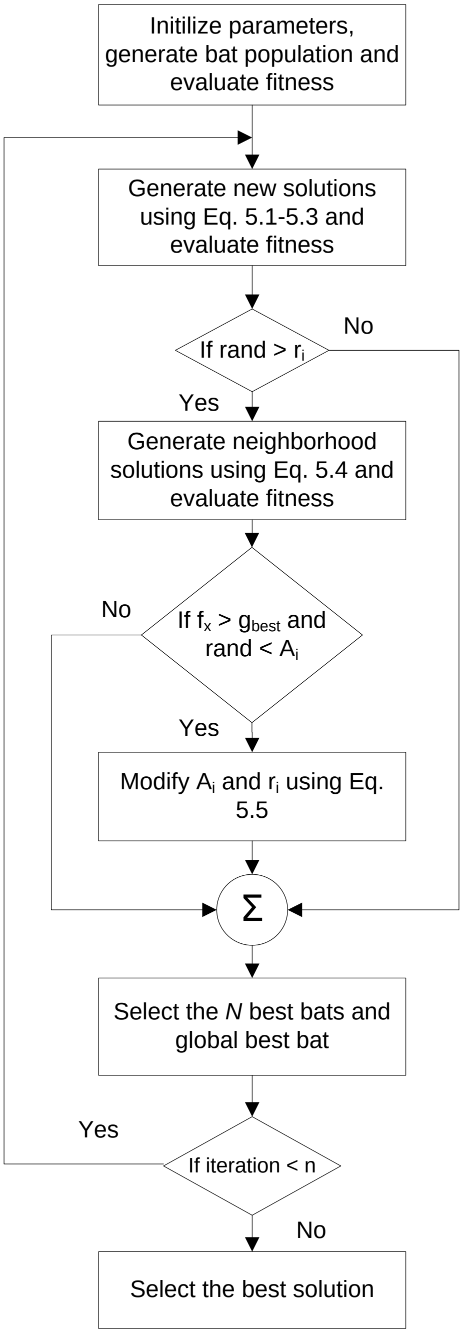

Initilize parameters, generate bat population and evaluate fitness
Generate new solutions using Eq. 5.1-5.3 and evaluate fitness
If rand > r_i
No
Yes
Generate neighborhood solutions using Eq. 5.4 and evaluate fitness
No
If $f_x > g_{best}$ and rand < A_i
Yes
Modify A_i and r_i using Eq. 5.5
Σ
Select the *N* best bats and global best bat
Yes
If iteration < n
No
Select the best solution

Figure 5.1: AB algorithm: Flow chart

5.2 Literature Review

Standard AB algorithm can provide very good convergence at an early stage by switching from exploration to exploitation. This makes it very attractive for classification, optimization, feature extraction, scheduling, data mining and others. However, a quick switching to exploitation stage, by varying A and r too quickly, leads to stagnation as the solution may be trapped in a local minima. In an effort to avoid this situation and enhance performance, different methods and strategies have been incorporated in AB algorithm to instil diversity of solution. Some notable variants of AB algorithm are:

Khan et al. [66] developed a fuzzy logic AB algorithm by redefining the velocity update equation, Equation 5.2, to update the fuzzy relationship between the components of the velocity vector. Komarasamy et al. [70] used K-means in AB algorithm for efficient clustering. Lin et al. [81] developed a chaotic AB algorithm using Lévy flights and chaotic maps. They used it to estimate the parameters of a dynamic biological system. Xie et al. [57] employed differential operator and Lévy flights in AB algorithm to solve function optimization problems. Zang et al. [158] used mutation to enhance the diversity of solution for their image matching problem. Wang et al. [159] incorporated harmony search to create a hybridized AB algorithm for numerical function optimization. Gandomi et al. [42] introduced chaos into AB algorithm to improve its global search ability. They proposed four different variants of chaotic AB algorithm with thirteen different chaotic maps and shown that some variants of chaotic AB algorithm can outperform standard AB algorithm.

5.3 Proposed Modified Artificial Bat Algorithm

The particle update procedure, after the neighborhood search, and loudness decrement steps of the original AB algorithm are modified in the proposed MAB algorithm.

The wavelength λ is chosen to vary instead of frequency f. λ and f are related by the following relation:

$$\lambda = \frac{v}{f} \tag{5.6}$$

where v is the velocity increment. The wavelength λ is allowed to vary in accordance with a maximum variance V_{max} in each time step in each dimension.

If a particle is selected for neighborhood search and its likelihood value is improved then its position is updated, contrary to the original algorithm where it is updated only if this solution is better than the global best solution. As a result the particles will start to converge towards the optimal best solution which will improve the ‘degeneracy’ issue. The values of A_i and r_i are also updated if the particle’s solution improves.

Moreover, in the original algorithm the value of A_i is decreased at a constant rate α if the solution improves, enhancing its probability to be selected for further modification. However, in the proposed algorithm, this value is varied according to the quality of the obtained solution:

$$A_i^{t+1} = 0.35e^{(0.9-J_k(i))}A_i^t \tag{5.7}$$

which clearly indicates that this factor will decrease abruptly if particle is in the vicinity of the optimal solution enhancing its chance to be selected for further modifications, and changes slowly if the improvement in the obtained solution is small. The pseudo code of the proposed MAB algorithm is given in Algorithm 2.

Algorithm 2 Modified artificial Bat algorithm

```
Randomly initialize bat population $x_i$ and $v_i$
Define pulse frequency $f_i$ for each bat $x_i$
Initialize pulse emission rate $r_i$ and loudness $A_i$
repeat
    for all bats $i$ do
        Generate new solutions using Equations 5.1-5.3
        Evaluate fitness of updated bats
        if $rand > r_i$ then Generate new solutions $\hat{x}_i$ using Equation 5.4 and Evaluate fitness;
        if $f(\hat{x}_i) < f(x_i)$ then $x_i = \hat{x}_i$ and $f(x_i) = f(\hat{x}_i)$;
        if $f(x_i) < f(g_{best})$ then $g_{best} = x_i$ and
            Modify $A_i$ using Equation 5.7 and $r_i$ using Equation 5.5;
    End for
    Select the $N$ best bats
until Stopping criteria
```

5.4 Experimental Results

5.4.1 Test Functions

Benchmark Test Suite

To evaluate the performance of the proposed algorithm, a selected set of standard benchmark functions are used. The benchmark functions have diversified properties in terms of modality, separability and valley landscape. The test suite considered in this work consists of six standard benchmark functions. These functions are as follow:

Rosenbrock's Function: also known as **Rosenbrock's Valley** or **Rosenbrock's Banana** function. It is a continuous, differentiable, non-separable, unimodal function. Its global minimum is located inside a long parabolic shaped flat valley. It is trivial to find the valley but convergence to the global minimum is difficult. It is defined as:

$$f_1(\boldsymbol{x}_i) = \sum_{i=1}^{d-1} 100(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i^2)^2 + (\boldsymbol{x}_i^2 - 1)^2 \qquad where - 15 \leq \boldsymbol{x}_i \leq 15 \tag{5.8}$$

It has a global minimum at $\boldsymbol{x}^* = (1, ..., 1)$ where the value of the function is $f(\boldsymbol{x}^*) = 0$.

Griewank's Function: is a continuous, differentiable, non-separable, multi-modal function. It has many widespread local minima which increases with dimensionality. Its aim is to test the recovery property of an optimization algorithm after failure. It is defined as:

$$f_2(\boldsymbol{x}_i) = -\prod_{i=1}^{d} \cos(\frac{\boldsymbol{x}_i}{\sqrt{i}}) + \sum_{i=1}^{d} \frac{\boldsymbol{x}_i^2}{4000} + 1 \qquad where - 600 \leq \boldsymbol{x}_i \leq 600 \tag{5.9}$$

Its global minimum is located at $\boldsymbol{x}^* = (0, ..., 0)$ and has a value $f(\boldsymbol{x}^*) = 0$.

Sphere Function: is a continuous, differentiable, separable, unimodal function. It has one global minima. It is of the form:

$$f_3(\boldsymbol{x}_i) = \sum_{i=1}^{d} \boldsymbol{x}_i^2 \qquad where - 5.12 \leq \boldsymbol{x}_i \leq 5.12 \tag{5.10}$$

It has a global minimum at $\boldsymbol{x}^* = (0, ..., 0)$ where the value of the function is $f(\boldsymbol{x}^*) = 0$.

Rastrigin's Function: is a non-convex, non-linear, multi-modal function. This function is fairly difficult to optimize due to its large search space and existence of a large number of local minima. These minima are induced by the addition of a cosine term to the sphere function. Its overall shape is flatter than Ackley's function which can complicate the general convergence towards the global optimum. Rastrigin's function is of the form:

$$f_4(\boldsymbol{x}_i) = d \times 10 + \sum_{i=1}^{d} (\boldsymbol{x}_i^2 - 10\cos(2\pi \boldsymbol{x}_i)) \qquad where - 15 \leq \boldsymbol{x}_i \leq 15 \tag{5.11}$$

It has a global minimum at $\boldsymbol{x}^* = (0, ..., 0)$ where the value of the function is $f(\boldsymbol{x}^*) = 0$.

Ackley's Function: is a continuous, differentiable, non-separable, multi-modal function. This function is moderately difficult to solve due to a single funnel like shape. Nevertheless, it has many local minima due to which a gradient based algorithm will trap in any of these minima. An optimization algorithm searching a wider solution space will achieve better results in this situation. It is of the form:

$$f_5(\boldsymbol{x}_i) = \sum_{i=1}^{d-1}(20+e-20e^{-0.2\sqrt{0.5(\boldsymbol{x}_{i+1}^2+\boldsymbol{x}_i^2)}}-e^{0.5(\cos(2\pi\boldsymbol{x}_{i+1})+\cos(2\pi\boldsymbol{x}_i))} \qquad where-32 \leq \boldsymbol{x}_i \leq 32 \tag{5.12}$$

Its global minimum is located at $\boldsymbol{x}^* = (0, ..., 0)$ and the function value at this point is $f(\boldsymbol{x}^*) = 0$.

Schaffer's F6 Function: has potential maxima that keeps on increasing as one gets closer to the global minimum. The optimization algorithm needs to overcome these gradually ascending peaks to reach the optimal solution. The Schaffer's F6 function is:

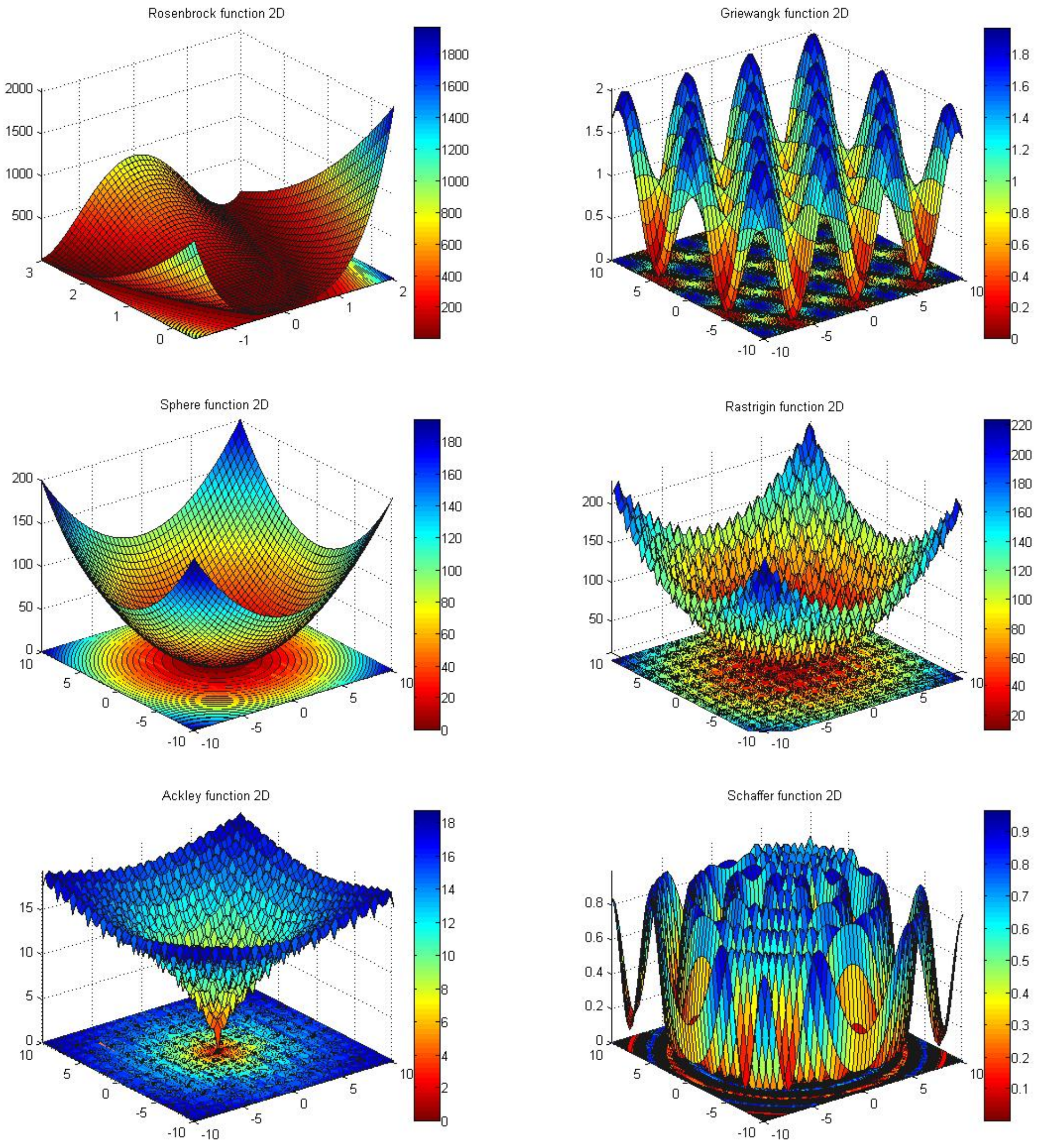
$$f_6(\boldsymbol{x}_i) = 0.5 + \frac{\sin^2(\sqrt{\boldsymbol{x}_i^2 + \boldsymbol{x}_{i+1}^2}) - 0.5}{[1 + 0.001(\boldsymbol{x}_i^2 + \boldsymbol{x}_{i+1}^2)]^2} \qquad where - 10 \leq \boldsymbol{x}_i \leq 10 \tag{5.13}$$

Its global minimum is located at $\boldsymbol{x}^* = (0, ..., 0)$ and the function value at this point is $f(\boldsymbol{x}^*) = 0$. The 2-dimensional form of each utilized benchmark test functions is shown in Figure 5.2.

Test Results

These benchmark test functions are used to compare the performance of the original AB algorithm and the proposed MAB algorithm. Since the dimensionality of the problem greatly affects the obtained results, the two algorithms are tested for problem dimensions of $d = 10$, $d = 20$ and $d = 30$. The size of the bat population is increased with dimension. For a problem dimension $d = 10$, both original AB and MAB algorithms are initialized with 250 bats and 20 generations are used. The bat population is doubled when the dimension is increased from $d = 10$ to $d = 20$ and tripled for dimension $d = 30$. Initial values of pulse emission rate and loudness are set to $r_0 = 0.5$ and $A_0 = 0.5$ respectively for AB algorithm. For MAB algorithm, pulse emission rate is the same as AB algorithm but loudness is varied according to Equation 5.7. The obtained minimum function values are compared in terms of best, mean and standard deviation measures in Table 5.1.

The results show that MAB algorithm has outperformed AB algorithm in all cases. The improvement in results is due to the fact that local evaluations are utilized to guide the local search. It is also noticed that, with the increase in dimension, the performance for both


Rosenbrock function 2D

Griewangk function 2D

Sphere function 2D

Rastrigin function 2D

Ackley function 2D

Schaffer function 2D

Figure 5.2: Benchmark test functions

routines deteriorates. The required number of particles to solve these types of optimization problems using population based optimization algorithms increases exponentially [87]. In our experiments, a limited number of particles are used and are linearly increasing with problem dimension but the built in exploration and enhanced exploitation characteristics of MAB algorithm allows it to approach a solution of these problems with a limited population of particles.

5.5 Conclusions

In this chapter, a MAB algorithm is proposed. AB algorithm is based on the echolocation behavior of bats. It combines the advantages of PSO and Harmony Search (HS). Hence, AB algorithm can be considered as a combination of PSO and local search. The population update and local search mechanism of AB algorithm is modified in the proposed algorithm. The performance of MAB algorithm is evaluated on a standard suite of benchmark optimization problems. MAB algorithm has outperformed original AB algorithm for all the test functions.

Table 5.1: Comparison of AB and MAB algorithms for increasing dimension benchmark function evaluation

Algo.	d	Meas.	f_1	f_2	f_3	f_4	f_5	f_6
AB	10	Best	1.13×10^{2}	3.18×10^{1}	3.11×10^{1}	7.13×10^{1}	1.58×10^{1}	1.81×10^{2}
		Mean	5.71×10^{2}	8.37×10^{1}	1.49×10^{2}	3.33×10^{2}	1.93×10^{1}	4.77×10^{2}
		STD	4.17×10^{1}	3.05×10^{1}	1.37×10^{1}	2.19×10^{1}	1.07×10^{1}	1.32×10^{1}
	20	Best	3.81×10^{2}	8.94×10^{1}	7.89×10^{1}	7.81×10^{1}	2.29×10^{2}	2.89×10^{2}
		Mean	5.99×10^{2}	1.47×10^{2}	3.52×10^{2}	3.55×10^{2}	3.47×10^{2}	8.14×10^{2}
		STD	4.87×10^{1}	1.64×10^{1}	4.59×10^{1}	2.51×10^{1}	1.88×10^{1}	2.17×10^{1}
	30	Best	4.93×10^{2}	2.15×10^{2}	1.78×10^{2}	9.17×10^{1}	3.81×10^{2}	6.17×10^{2}
		Mean	7.81×10^{2}	4.51×10^{2}	4.33×10^{2}	4.21×10^{2}	4.93×10^{2}	9.11×10^{2}
		STD	2.78×10^{1}	1.61×10^{1}	1.88×10^{1}	1.96×10^{1}	1.87×10^{1}	3.11×10^{1}
MAB	10	Best	5.77×10^{-2}	3.71×10^{-7}	1.37×10^{-8}	6.47×10^{0}	7.91×10^{-3}	2.78×10^{-2}
		Mean	6.17×10^{1}	3.78×10^{-6}	1.75×10^{-4}	2.11×10^{1}	1.71×10^{1}	4.81×10^{-1}
		STD	7.99×10^{0}	1.19×10^{-7}	4.77×10^{-6}	1.66×10^{1}	1.23×10^{1}	6.37×10^{0}
	20	Best	1.18×10^{-1}	4.93×10^{-6}	5.78×10^{-4}	1.89×10^{-1}	3.79×10^{-4}	4.97×10^{-2}
		Mean	2.37×10^{-1}	8.76×10^{-6}	6.42×10^{-3}	2.71×10^{1}	4.89×10^{-2}	9.83×10^{-2}
		STD	3.11×10^{0}	2.17×10^{-7}	1.99×10^{-2}	6.69×10^{0}	1.81×10^{-1}	9.99×10^{-1}
	30	Best	9.11×10^{0}	5.47×10^{-5}	4.71×10^{-1}	1.93×10^{0}	6.14×10^{-3}	1.81×10^{-1}
		Mean	6.95×10^{1}	9.11×10^{-5}	3.85×10^{0}	2.11×10^{1}	2.77×10^{-2}	5.67×10^{0}
		STD	4.23×10^{1}	3.12×10^{-6}	1.77×10^{1}	4.73×10^{0}	3.17×10^{-1}	1.78×10^{0}

Chapter 6

Differential Mutated Artificial Immune System Algorithm

In this chapter, a bio-inspired Artificial Immune System (AIS) algorithm is proposed. Section 6.1 provides a background on the natural immune system. It covers the main actors involved in the immune system and how they work together to provide a defence mechanism against invaders. Section 6.2, briefly describes the clonal selection theory which is the basis of the AIS algorithm developed in this chapter. Section 6.3 depicts how the natural immune system is modified to solve engineering optimization problems. Section 6.4 reviews some of the key developments in AIS research. Section 6.5 explains the proposed differential evolution based AIS algorithm. Experimental results are given in Section 6.6. The chapter is concluded in Section 6.7 with a discussion of the obtained results.

6.1 Natural Immune System

Immune mechanism is a powerful source of inspiration for the development of computational tools. Research interest in AIS field increased immensely among scientists, engineers and mathematicians in the last decade [144]. The immune system is a defence system that evolves to protect from harmful micro-organisms, such as bacteria, viruses and parasites, collectively called pathogens. Immune system consists of cells, molecules and organs that are capable of identifying and combating the exogenous infectious micro-organisms from own body cells. The natural immune system and immune-inspired artificial systems are generally understood to exhibit learning, memory, decentralization, distribution, collaboration and imperfect recognition mechanisms.

The natural immune system has a multi-layered architecture. Skin and the respiratory system act as the first barrier against invaders. Saliva, sweat, tears and stomach acid provides a physiological barrier as they contain destructive enzymes to counteract the invaders. Innate and adaptive immune systems are the next two layers in this defence scheme. These components are shown in Figure 6.1.

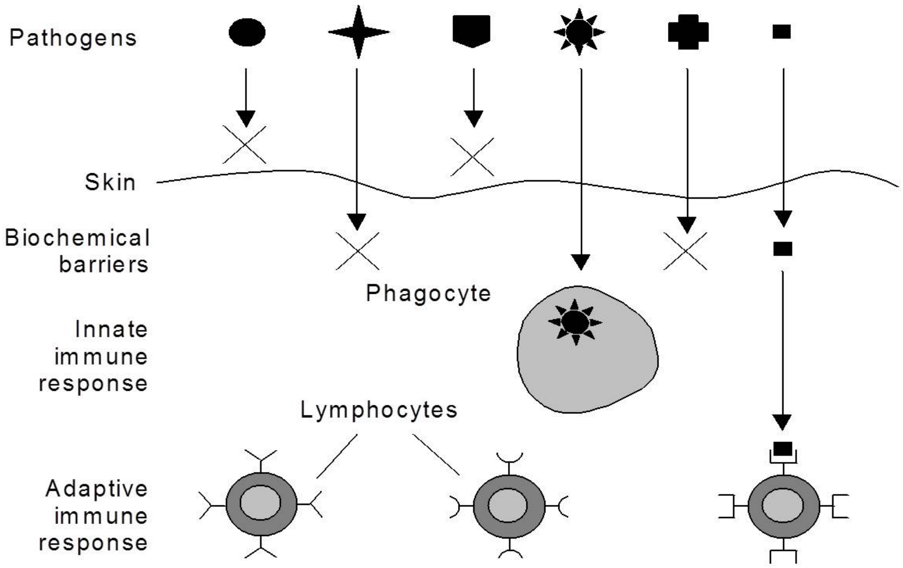


Figure 6.1: Human immune system response to pathogens [32]

Innate immunity provides a general defence against the pathogens. They are encoded in the gene of the species and do not evolve. They have an inborn ability to recognize certain microbes and immediately destroy them. The innate system also induces the expression of co-stimulatory signals that activates the T-cells which initiates the adaptive immune response. The adaptive immune system is directed towards specific antigens. It mainly consists of lymphocytes, also called B-cells and T-cells. These immune agents are modified by exposure to invaders. The innate and adaptive immune systems operate on different time scales. The innate immune system initializes its action either immediately or within a few minutes of antigen exposure. Whereas, the adaptive immune system takes days to initiate a reaction. The combination of these two provide an effective immune system.

The tissues and organs that are part of the immune system are termed as lymphoid organs and are distributed throughout the body. Tonsils, adenoids, lymphatic vessels, bone marrow, lymph nodes, thymus, spleen and appendix are part of the lymphoid organs. A variety of immune cells are originated and matured in bone marrow. They are then transformed to the patrolling tissues, circulating in the blood and lymphatic vessels. Some of them provide general defence while others are trained to combat specific pathogens. Upon activation, B-cells differentiate to produce plasma cells and secrete Antibodies (Ab) in response of exogenous protein like bacteria, viruses and tumor cells. Each type of B-cell produces a specific antibody. Each antibody binds to a particular protein, which serves as a way to signal other cells to kill, ingest or remove the bounded substance. The T-cells are matured within thymus. They regulate other cell actions and directly attack the host infected cells.

How the Immune System Works?

This army of cells and molecules works together to target Antigens (Ag) which are foreign molecules, bacterium or any other invader. Specialized antigen cells are present in the body that ingest and digest the antigens and fragment them into antigenic peptides [100]. Pieces of these peptides are joined to Major Histocompatibility Complex (MHC) molecules which attach to the cell surface. T-cells have receptor molecules to recognize different peptide-MHC combinations. T-cells, activated by this recognition, divide and secrete a chemical signal, lymphokines, that mobilizes other components of the immune system. The B-cells, that have receptor molecules for that specific signal, respond to these signals. B-cells can also recognize the parts of the antigen in free solution without MHC molecules. The activated B-cells divide and differentiate into plasma cells and secrete antibody proteins which are a soluble form of receptors. These antibodies bind with antigen to neutralize them or precipitate their destruction by complement enzymes or by scavenging cells. Some T-cells and B-cells become memory cells to eliminate the same antigen, if found in future. Since antibodies in B-cells undergo mutation and editing after repeated immunization, their response improves and this phenomenon is termed as affinity maturation.

6.2 Clonal Selection Theory of Immunology

The body contains millions of lymphocytes with different specificities to form a repertoire of individuals, called B-cell and T-cell. In adaptive immune response, this repertoire undergoes a selection mechanism. This selection mechanism is based on the established clonal selection theory proposed by Burnet [20] in 1959. It explains how an adequate number of lymphocytes are produced which are capable of recognizing a particular pathogen and combating it.

The theory states that activation of lymphocytes occur when there is sufficient antigen binding. The activated lymphocyte undergoes cloning, thus an expression of the original lymphocyte takes place. The theory is based on four principles:

1. Each lymphocyte bears a single receptor with unique specificity.
2. Lymphocyte activation occurs when there is high affinity between lymphocyte and antigen.
3. The activated lymphocyte under go differentiation and derived cells bear the same specificity as parent.
4. Lymphocytes having receptors specific to host molecules are deleted.

Essentially, clonal selection theory is based on cloning and affinity maturation concept. When the body is exposed to an antigen, the B-cells with the best binding or affinity to the antigen proliferate by cloning. These cloned cells are called antibodies. The cells with higher affinity to antigen survive. Some of the cells with higher affinity become memory cells. The clonal selection mechanism is sketched in Figure 6.2.

6.3 Artificial Immune System

AIS algorithms are considered highly robust, adaptive, self-organized and inherently parallel structured [144]. It has powerful learning capabilities and depicts an evolutionary response. It has the ability to escape the local optimum region through mutation and strong local search capability through cloning. It also adds diversification by replacing the worst performing individuals in the population.

In AIS terminology, the optimization problem to be solved is the antigen, generated solutions are the antibodies, fitness value (objective function evaluation) is the affinity, cloning is the reproduction of solutions, mutation is the random modification of solutions, and receptor editing is the diversification of solutions.

In the AIS variant inspired by the clonal selection theory [32], a population of antibodies (solutions) is randomly generated. On the basis of antigen affinity, N antibodies are

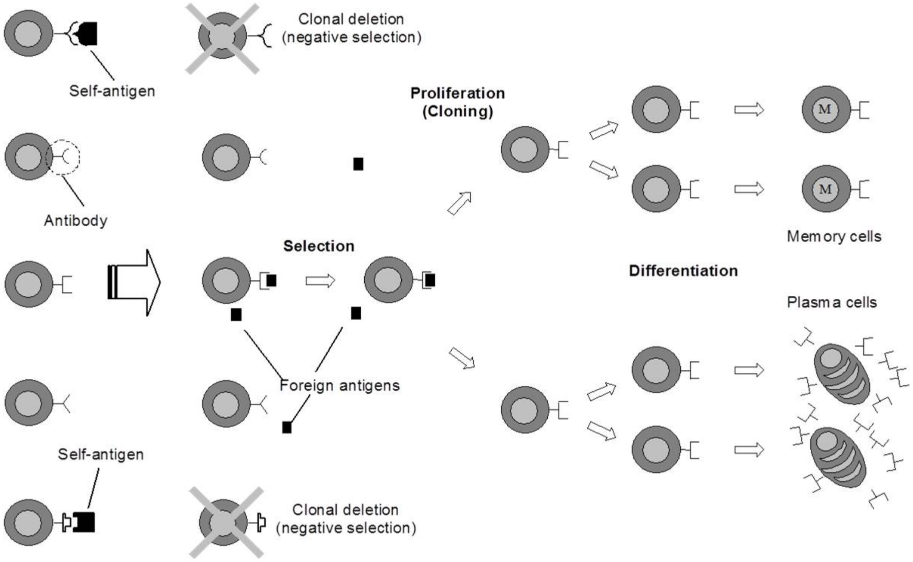


Figure 6.2: Negative selection, proliferation and differentiation in clonal selection theory [32]

selected using a selection criteria, such as elitism, roulette wheel or tournament selection. These selected antibodies go through a cloning operation. The cloning process forms the local search tool of the algorithm. The antibodies with higher affinity generate more clones. Then a subset of the cloned antibodies undergoes hyper mutation or multiple mutation and receptor editing or diversification operations. This diversification process forms the basis of the global search mechanism of the algorithm. These operations lend them the ability to escape local optima. The mutation rate is kept inversely proportional to antigen affinity. Some of the best members are added to the memory cells. A percentage of the worst members of the previous population of antibodies is replaced with some randomly generated new solutions which will add diversity to the population. The block diagram in Figure 6.3 depicts the major steps of the AIS algorithm.

6.4 Recent Research Contributions in AIS

A plethora of AIS models and techniques are reported in literature based on different immune theories but this discussion is limited to algorithms imitating the clonal selection theory. A clonal selection based algorithm, CLONALG, was proposed by de Castro et al.

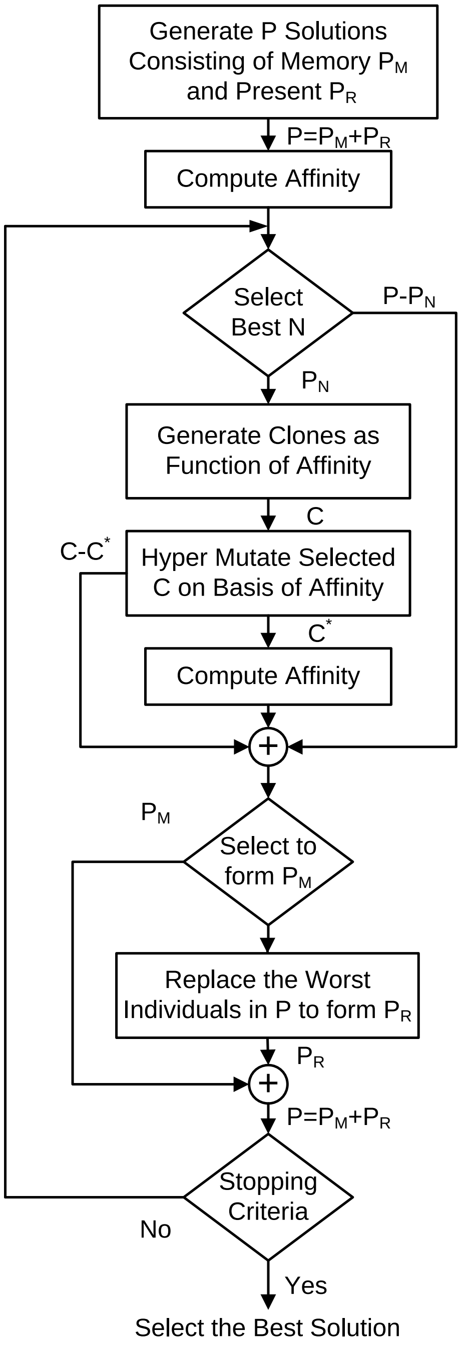

Generate P Solutions Consisting of Memory P_M and Present P_R
$P=P_M+P_R$
Compute Affinity
Select Best N
$P-P_N$
P_N
Generate Clones as Function of Affinity
C
$C-C^*$
Hyper Mutate Selected C on Basis of Affinity
C^*
Compute Affinity
+
P_M
Select to form P_M
Replace the Worst Individuals in P to form P_R
P_R
+
$P=P_M+P_R$
Stopping Criteria
No
Yes
Select the Best Solution

Figure 6.3: Schematic of clonal selection based AIS algorithm

[32]. The algorithm randomly generates N solutions of the optimization problem. Best antibodies go through cloning and mutation process to construct new candidate solutions. These solutions are evaluated and a percentage of the best solutions is added to the population. Further, a percentage of worst antibodies are discarded and replaced with new randomly created solutions.

Rouchen et al. [120] used immunity monoclonal and immunity polyclonal strategy algorithms to develop their optimization scheme. They used it to solve multi-objective optimization problem. Zuo et al. [108] used chaotic variables instead of cloning to perform local search in their algorithm.

Garrett et al. [43] proposed a method for selecting the amount of mutation and the number of clones in the original CLONALG algorithm. Rouchen et al. [82] proposed an adaptive immune clonal strategy algorithm and evaluated its performance on benchmark numerical optimization problems. It integrates local search with global search. The algorithm dynamically assigns antibodies to the immune memory unit and antibody population, on the basis of Ab-Ab and Ab-Ag affinities.

Yu et al. [171] introduced a learning operator to enhance the learning mechanism of CLONALG. Cutello et al. [26] proposed an algorithm that included a cloning operator that explores the neighborhood at each point in the search space. They have also introduced an aging operator that removes oldest candidate solutions to introduce diversity and avoid local minima. They have further developed an algorithm where real coded variables are used and introduced a new hypermutation operator inversely proportional to affinity [27].

Gong et al. [47] used self-adaptive chaotic mutation. The proposed algorithm generates initial antibody population and adopts the logistics chaotic sequence to introduce mutations and hypermutation. They later proposed a differential evolution based clonal selection algorithm which combines the clonal selection theory and differential evolution theory [48].

Dabrowski et al. [28] proposed a parallel clonal selection algorithm to solve a graph coloring problem. It uses an island model where every processor works independently with its own pool of antibodies to improve performance. Lu et al. [84] developed clonal chaos adjustment algorithm for multimodal function optimization. It uses ergodic and dynamic properties of a chaotic system to enhance global convergence and introduces a chaotic search mechanism to improve search efficiency.

6.5 Proposed Differential Mutated AIS Algorithm

In the proposed DM-AIS algorithm, a two stage mutation operation is specifically devised to drive global and local search operations. The first stage of mutation is designed to add more diversity to the population, thus, carry out a more through exploration of the solution space. The second stage of mutation is geared towards local search. Differential mutation operation is proposed to add variations to the two mutation stages. A parent fitness based adaptive scalar mutation factor F is suggested that automatically switches the search between global and local. Further, the variation among the clones is generated in proportional to the maximum variance allowed in each time step in each problem dimension. The cloning and mutation operations are carried out as follow.

Cloning Step

In strictly biological terms, *cloning* means creating an exact copy of an organism, cell or DNA sequence. However, in evolutionary computation, clones are slightly variant species of their parents. Antibodies are selected on the basis of their affinity for cloning process. The Roulette wheel selection method is used here for parent antibodies selection. This method gives a better chance of selection to the antibodies with greater affinity. However, with this fitness proportionate selection, some weaker antibodies may also survive the selection procedure. The selection of weak antibodies has some advantages, as some weak solutions may include components useful in mutation step. The clones are expressed as:

$$x_k^{offspring,l} = x_k^{parent,i} + \alpha \times V_{max} \tag{6.1}$$

where $x_k^{parent,i}$ represents the i^{th} parent at time step k, and $x_k^{offspring,l}$ represents the l^{th} offspring of the i^{th} parent at time step k. The parameter α, controls the dispersion of the offsprings from their parents and V_{max} is the maximum variance allowed in each time step in each dimension.

Mutation Step

In biological terms *mutation* means a sudden change in gene characteristics of a chromosome. In evolutionary computation field, it represents a random change or perturbation of a solution. In differential mutation, a part of the Differential Evolution (DE) technique [141, 140] used for continuous optimization problems, a weighted difference between two randomly selected individuals is the source of random variation. The offspring is obtained by adding the mutant vector to the parent vector as:

$$x_k^{offspring,i} = x_k^{parent,i} + F \times (x_k^j - x_k^m) \tag{6.2}$$

The mutant vector is scaled by a scalar factor F, which typically lies in the interval $[0.4 - 1]$. The process is graphically shown in Figure 6.4 for a simple 2-dimensional differential mutation.

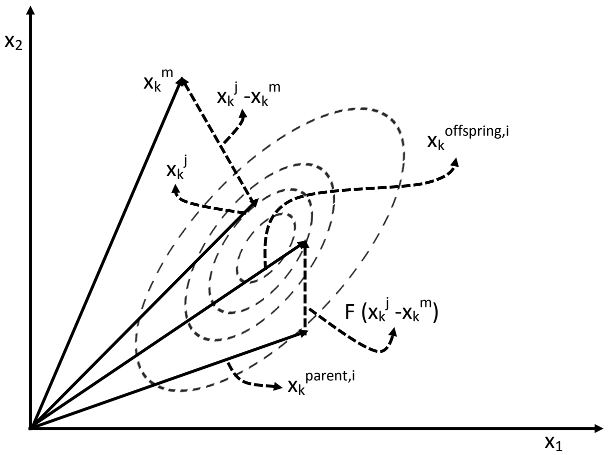


Figure 6.4: Differential mutation scheme for 2-D antibodies [29]

An adaptation rule based on the fitness value of the selected parent is proposed to determine the scale factor F. This method favors a strong local search, if the fitness value is high, and on the contrary a global search, if the fitness value of the selected parent is low. The fitness value of the parent antibody is weighted by an exponential function to emphasis extreme values for exploitation and exploration as:

$$F = e^{-5 \times fitness} \tag{6.3}$$

Figure 6.5 shows how the factor F varies as the fitness value changes.

Furthermore, a primary and secondary mutation operations are proposed and different mutation strategies are adapted for each of them. During primary mutation, antibodies are selected randomly without considering fitness values. Here, the two antibodies, use to generate a differential mutant are also randomly selected. This will enhance the probability to search a larger solution space as more changes in the offsprings may occur. At this stage a one-to-one competition between parent and offspring antibodies is used for survival. If

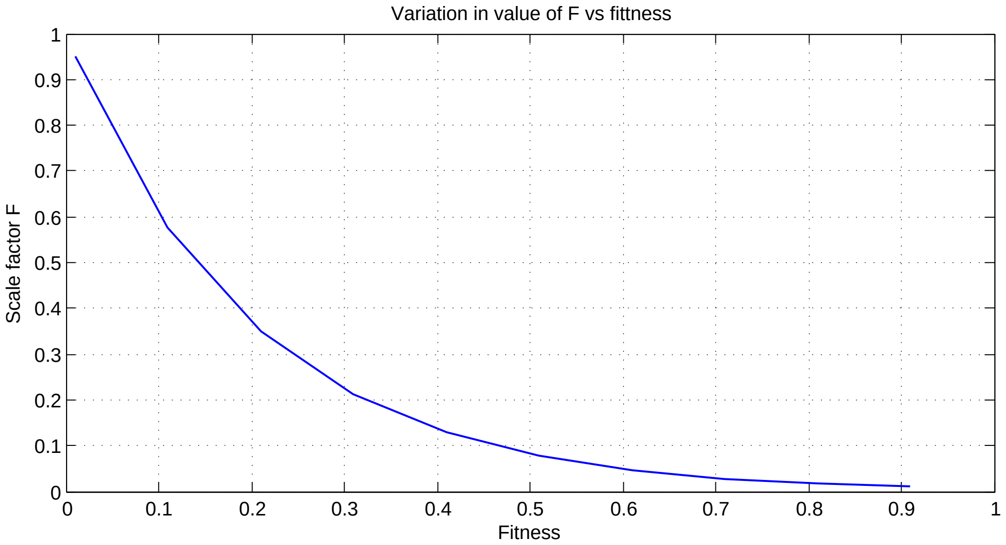


Figure 6.5: Variation in scale factor F as a function of fitness value

the fitness value of the offspring is improved, then the parent is replaced by it, thereby ensuring that the best performing individuals in a certain neighborhood in the search space survive. This type of parent-offspring competition provides better diversity as compared to ranking or tournament based selection, where the elites and their offsprings may dominate the solution and give rise to the 'degeneracy' problem associated with generic PF.

Secondary mutation occurs as a part of the cloning process. Following cloning, The Roulette wheel selection method is used to select a parent vector for mutation. The two immediate neighbors of the selected clone are used for differential mutation of that clone. It is evident that this scheme generally favors exploitation as the solution is based on nearest neighbors.

To maintain a constant population size, antibodies are selected from the whole population of antibodies. Some of the best antibodies are added to the memory bank to enrich it for future instances. Some of the worst antibodies are discarded and replaced by randomly generated antibodies to maintain the diversity of the solution.

The block diagram in Figure 6.6 delineates the steps involved in the proposed DM-AIS algorithm and the pseudocode is given in Algorithm 3.

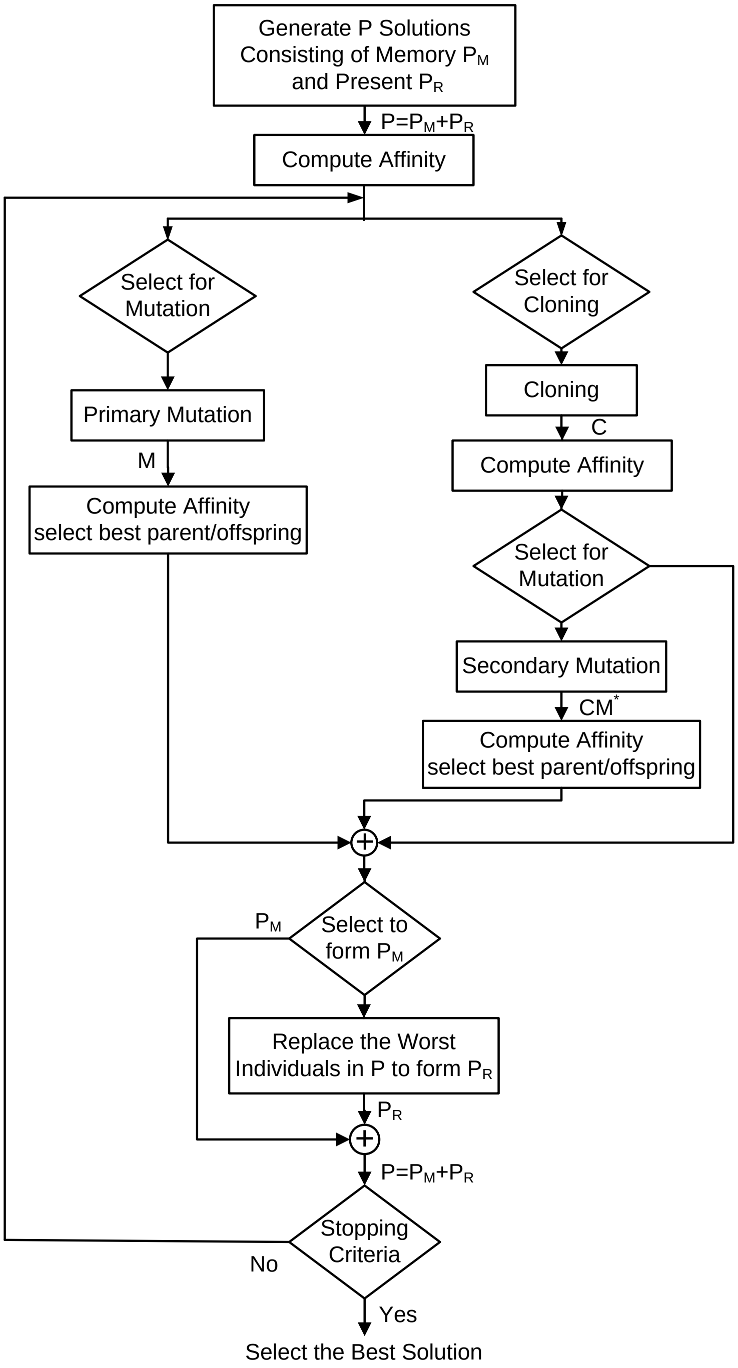


Figure 6.6: Schematic of the proposed DM-AIS algorithm

Algorithm 3 Differential mutated artificial immune system algorithm

```
Generate initial population of antibodies consisting of memory $P_M$ and present $P_R$
Compute affinity
repeat
    Select $N_1$ antibodies randomly for mutation                ▷ Primary mutation
    Perform primary mutation for the selected antibodies
    Compute affinity and Select best among parent/offspring

    Select $N_2$ antibodies on affinity basis for cloning        ▷ Cloning
    Clone the selected antibodies and Compute affinity
    Select $N_3$ antibodies from clones for mutation             ▷ Secondary mutation
    Perform secondary mutation for the selected antibodies
    Compute affinity

    Add best antibodies to $P_M$                                 ▷ Update of memory bank
    Remove worst members of the antibodies pool
    Add new random antibodies into the population
until Stopping criteria
```

6.6 Experimental Results: DM-AIS for Ball Tracking

To evaluate the efficacy of the proposed DM-AIS algorithm, a low-dimensional object tracking problem, where video sequence of a ball bouncing on the surface of a table, is a considered. The video is captured at 15 frames per second and each frame consists of 240x320 pixels. A total of 60 frames are considered in this experiment. Background subtraction is used to extract the ball position in the video frame. Then, the centroid of this detected ball region is computed which is later used as the desired ball position. The dynamics of the bouncing ball are described by a set of states $\boldsymbol{x}_k$. A constant velocity temporal model is used to predict the ball position in the next video frame. The position and velocities in x and y directions are the states of the system.

A generic PF algorithm with 250 particles and a PSO algorithm with 50 particles and 5-iterations of PSO run are used. For the DM-AIS algorithm, 25 antibodies are used in the initial population, 5 antibodies are selected for primary mutation, and 5 antibodies are selected for cloning and 30 clones are produced. From these 30-clones, 5 are selected for secondary mutation. The DM-AIS algorithm is also run for 5 iterations. This setup make an equal number of function evaluations, 250 function evaluations, for the three algorithms.

The tracking results of this experimentation are listed in Table 6.1. It is clear that the generic PF tracking results are inferior to the other two algorithms. The reason for this poor performance is the absence of a local search mechanism in PF algorithm. The performance of DM-AIS is the best, owing to the strong local search capability due to cloning and secondary mutation step, and the existence of a global search ability due to primary mutation, which is beneficial for locating the ball when its direction of motion changes.

Selected frames are shown in Figure 6.7. These frames are selected as a change in direction of vertical motion takes place, namely when the ball bounces off the table or when it starts descending from the top of its bounce. Here, the red circles represent the observed ball positions, obtained by background subtraction and the green circles represent the predicted ball positions. It is also evident that DM-AIS is the most successful algorithm, both the observed (red) and predicted (green) ball positions are superimposed on each other for DM-AIS algorithm.

Table 6.1: Comparison of PF, PSO, DM-AIS MSE in ball tracking

Algorithm	Particles	Iterations	Function Eval	MSE/frame (in pixels)
PF	250	1	250	1.1925
PSO	50	5	250	0.4283
DM-AIS	25	5	250	0.0891

6.7 Conclusions

In this chapter, a new AIS based optimization algorithm, DM-AIS, is proposed. The algorithm is based on two mutation operations. In primary mutation, the parent antibodies and the two antibodies used to generate a differential mutant are randomly selected, thereby driving the algorithm for exploration of the solution space. In secondary mutation, the parent antibodies are selected from clones and nearest neighbors are utilized to generate differential mutant vectors. This process is geared more towards local search. Furthermore, the scalar mutation factor F is adaptively updated based on the parent fitness value. In this way, if the fitness value is high, it automatically favors local search. On the other hand, if the fitness is low, it spreads the mutated antibodies for global search. The proposed optimization scheme is evaluated for a low-dimensional video tracking problem and shown to outperform the other algorithms considered in this comparative study.

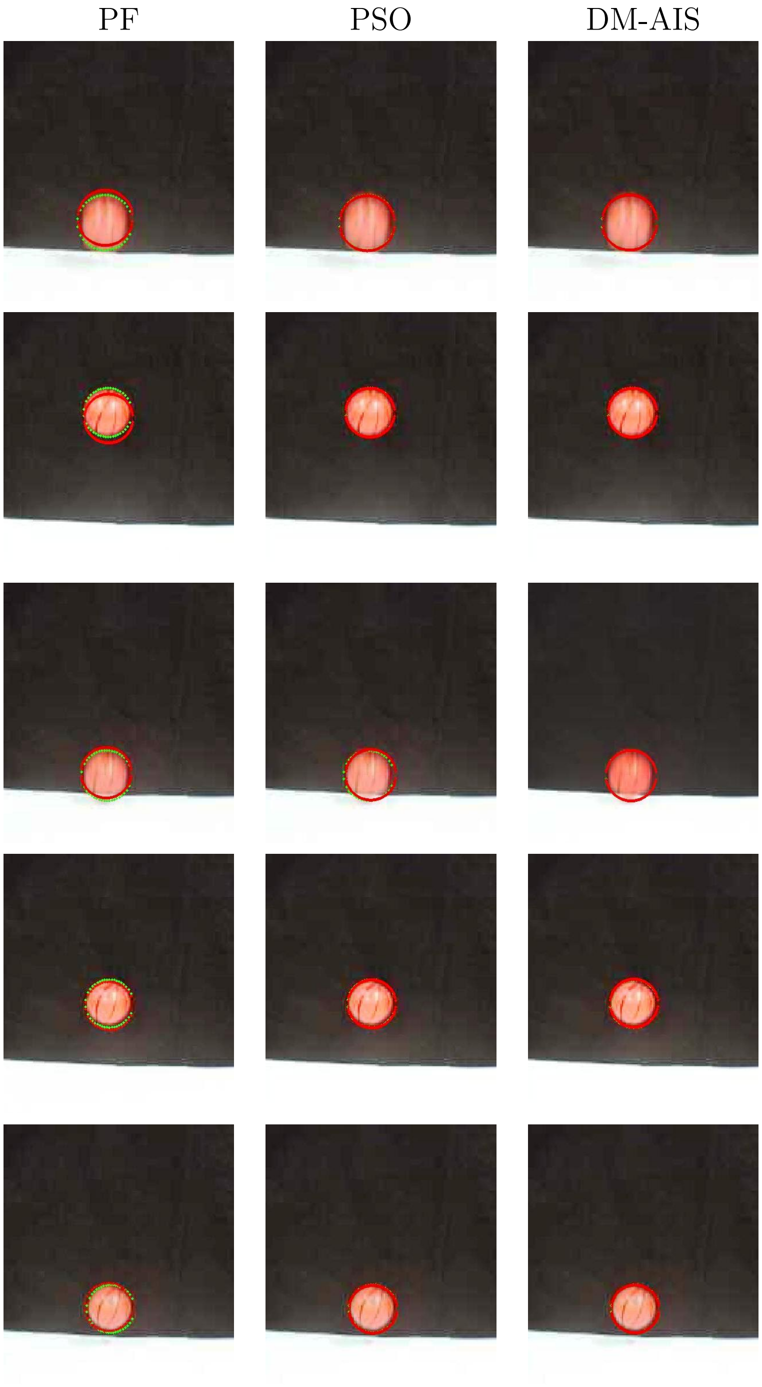


Figure 6.7: Tracking a bouncing ball, Frame# 14, 19, 27, 31 and 38

Chapter 7

A Hybrid Evolutionary Algorithm

Hybrid algorithms have become popular for solving complex and intricate problems. A hybrid optimization algorithm based on PSO and AIS is proposed in this chapter. Section 7.1 explains the reasons for hybridization. Section 7.2 investigates the strengths and weaknesses of PSO and AIS algorithms used for hybridization. Section 7.3 gives literature review of the state-of-the-art in the field of PSO-AIS hybridization. Section 7.4 describes the proposed hybrid scheme. Section 7.5 is dedicated to experimental results and Section 7.6 discusses the obtained results.

7.1 Hybridization

Use of hybrid algorithms is a new and successful trend in optimization. The main motivation is to obtain a better performing algorithm that combines the advantages of individual algorithms. Hybrid algorithms benefit from synergy. However, choosing an adequate combination of component algorithms to achieve better overall performance in a particular situation or problem is not trivial.

In particular, the hybridization of a population based metaheuristic with a local search method is extensively explored for optimization [16]. Generally, at the start of an optimization strategy, a global exploration of the search space is required to probe a wider region for the perspective solution and later on successively drive to more promising regions. The strength of the population based methods is their exploration capability. On the other hand, local search strategies have the capability of searching for a better solution in the vicinity of a given starting solution. Therefore, population based methods can quickly identify promising areas in the search space then local search methods can quickly determine the best solution within those areas. Such a synergistic use of population-based exploration and local search-based exploitation sounds very promising.

Indeed, hybrid algorithms have been applied to solve a diverse set of problems including medical image processing, engineering design problems, scheduling, data mining, unconstrained optimization problems etc.

7.2 Rationale for PSO and AIS Hybridization

Both PSO and AIS are optimization techniques inspired by nature. There are plethora of successful applications of both to solve numerical as well as real world optimization problems. However, despite their common properties of robustness, learning, decentralized and distributive nature neither of them is a panacea. Each of these algorithms has some advantages which are useful for certain aspects of optimization.

PSO is based on information sharing among the particles and memory effect due to retention of the most promising position of each particle. The most optimistic, global best, particle transmit information to other particles. PSO is computationally very simple as compared to evolutionary algorithms which are generally based on mutation, cloning and crossover operations to create diversity. There are only a few parameters (inertial weight, cognitive learning factor and social learning factor) that need to be tuned. It is based on real coded problem parameters which are equal to the dimension of the problem. It has also been demonstrated that PSO is faster and computationally less expensive than GA or Simulated Annealing (SA) [147].

One of the main advantage of PSO, rapid convergence, can also be its main weakness, as premature convergence to a suboptimal solution may stagnate the swarm without any pressure to continue exploration. Low fitness level of particles can be a good indication of stagnation. The PSO algorithm also has a poor local search ability. Due to random nature of velocity update, PSO algorithm has found it difficult to converge to optimal solution as the particles arrive in the vicinity of the optimal solution, hence shows slow convergence in the refined search stage.

On the other hand, AIS is a self-organizing, self-learning highly robust optimization algorithm. It does not require optimization of any system parameters. The strength of AIS algorithm is its parallel search nature. In AIS algorithm, mutation and cloning are the two aspects of exploration and exploitation. Mutation operation generates new solutions which help the algorithm to avoid trapping in local minima and cloning operation lends strong local search ability and aids in searching for the optimal solution which aids algorithm's likely converge towards high quality solutions. However, the rate of convergence for AIS is slow as compared to other evolutionary algorithms, such as GA. This can be the result of the absence of crossover operation which is part of other evolutionary algorithms.

7.3 Recent Research Contributions

There are a number of successful hybridizations of PSO algorithm with ant colony optimization, genetic algorithm, simulated annealing, differential evolution, tabu search, fuzzy logic, gradient descent, Nelder Mead simplex method to name a few [147]. This discussion is limited to those key hybrid algorithms where PSO and AIS have been used [45, 162, 85, 80, 46, 79, 164, 167].

Ge et al. [45] proposed an Immune Particle Swarm Optimization (IPSO) based on receptor editing. The method works by editing the low affinity particles after each generation. The proposed algorithm was used to train HMMs, which were later integrated with IPSO.

Wang et al. [162] used PSO algorithm to improve the antibodies before the cloning operation. Later on, this hybrid algorithm was used for unidimensional and multidimensional function optimization. Lv et al. [85] combined information diffusion and clonal selection to achieve a diversified search mechanism in PSO and avoided being trapped in local optimal solutions. Cheng-Jian et al. [80] employed PSO to improve the mutation scheme. The mutation step is replaced by a single iteration of PSO. They claimed better performance of their algorithm on synthetic and real data sets.

Ge et al. [46] proposed a hybrid of PSO and AIS algorithms for job shop scheduling problem. In the PSO algorithm a novel concept for the distance and velocity of a particle is presented to pave the way for the job shop scheduling problem and models of vaccination and receptor editing are designed to improve the immune performance.

Li et al. [79] proposed immune simplex PSO algorithm for numerical optimization problem. Orthogonality principle is used for even distribution of initial population. Then cloning and mutation operations are applied and finally the velocities of particles are modified, as they tend to become zero close to the optimal solution, by resetting the position according to simplex principle.

Yap et al. [170] proposed a Hybrid AIS (HAIS) algorithm that attempts to combine the good features of AIS and PSO in order to improve the convergence of AIS. The best half of the PSO population and a randomly initialized half of the population were used as the initial antibodies of AIS algorithm. This whole set of antibodies is cloned and mutated to search for the optimal solution.

Yu [164] proposed a hybrid algorithm using Artificial Immune Algorithm (AIA) and PSO algorithm to solve unconstrained global optimization problems. Using an external AIA, constriction coefficient, cognitive parameter and social parameter of PSO are optimized. The internal PSO algorithm is then used to solve the optimization problem.

Qui [167] proposed an algorithm in which clonal selection and immune network theories of artificial immune system are integrated with PSO. The clonal selection theory builds up the framework of the system and immune network theory is applied to increase the diversity of antibodies. Later on, a modified version of PSO is used to hypermutate the antibodies to accelerate the search procedure.

7.4 Proposed Hybrid PSO-AIS Algorithm

In order to alleviate the drawbacks of both PSO and AIS algorithms a new hybrid algorithm, PSO accelerated AIS (PSO-AIS) algorithm, is proposed. The proposed PSO-AIS algorithm, initializes with PSO algorithm which runs for a few iterations to drive the swarm towards the global best particle. These particles act as the initial population of antibodies for the AIS algorithm which has a strong local search mechanism.

The AP-PSO algorithm described in Section 4.2.1 is used as the primary optimizer in this work. This optimizer works as a global search algorithm and due to its fast initial convergence property, quickly drives the particles towards the more promising regions of the search space. It is chosen as a global explorer as it performs better than MAB algorithm, the other global search algorithm developed in this work. The DM-AIS algorithm proposed in Section 6.5 and depicted in Figure 6.6 is used to perform the local search. The proposed algorithm involves a novel two stage mutation operation to improve the efficacy of the AP-PSO optimized antibodies. The primary mutation is designed to improve the global search capability, while the secondary mutation is designed as to be biased towards local search. The block diagram in Figure 7.1 delineates the steps of this hybrid algorithm and its pseudocode given in Algorithm 4.

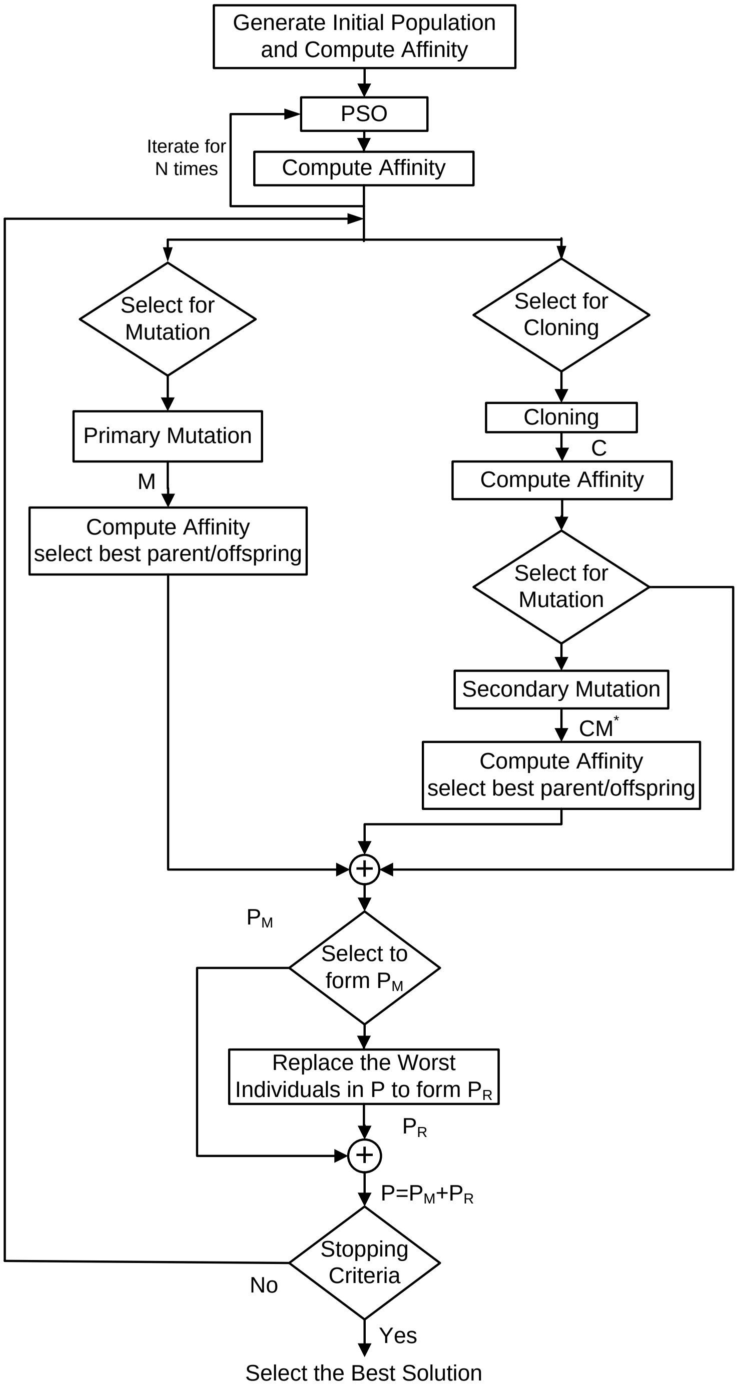


Figure 7.1: Block diagram of PSO-AIS algorithm

Algorithm 4 Hybrid PSO-AIS algorithm

```
Randomly initialize $x_i$ and $v_i$                                   ▷ AP-PSO Stage
Evaluate fitness $f(x_i)$
Set $p_{best} = x_i$ and select $g_{best}$
repeat
    for all particles $i$ do
        Update parameters $w$, $c_1$ and $c_2$ using Equations 4.3-4.5
        Update Velocities and positions using Equations 4.1-4.2
        Evaluate fitness of updated particles
        if $f(x_i) < f(p_{best}i)$ then $p_{best}i = x_i$;
        if $f(x_i) < f(g_{best})$ then $g_{best} = x_i$;
        Update $x_i$ and $v_i$
    End for
until Stopping criteria

            ▷ Final particles of AP-PSO stage act as initial antibodies for DM-AIS stage

repeat                                                                ▷ DM-AIS Stage
    Select $N_1$ antibodies randomly for mutation                     ▷ Primary mutation
    Perform primary mutation for the selected antibodies
    Compute affinity and Select best among parent/offspring

    Select $N_2$ antibodies on affinity basis for cloning             ▷ Cloning
    Clone the selected antibodies and Compute affinity
    Select $N_3$ antibodies from clones for mutation                  ▷ Secondary mutation
    Perform secondary mutation for the selected antibodies
    Compute affinity

    Add best antibodies to $P_M$                                      ▷ Update of memory bank
    Remove worst members of the antibodies pool
    Add new random antibodies into the population
until Stopping criteria
```

7.5 Simulation and Experimental Results

The performance of the proposed PSO-AIS algorithm is compared with PF, APF and PSO algorithms. Experiments are conducted to track the output of a non-stationary univariate growth model and on a vehicle tracking problem where abrupt jumps in the output are induced.

7.5.1 Univariate Growth Model

The univariate model is popular in economics. It is a non-stationary, highly non-linear model and is also bimodal in nature. It is used for performance analysis of estimation methods. It is defined as:

$$x_k = c_1 \cdot x_{k-1} + c_2 \cdot \frac{x_{k-1}}{1 - x_{k-1}^2} + c_3 \cdot \cos(1.2(k-1)) + w_k \tag{7.1}$$

$$y_k = \frac{x_k^2}{20} + v_k \tag{7.2}$$

where $w_k \sim N(0, \sigma_w^2)$ and $v_k \sim N(0, \sigma_v^2)$ are mutually independent Gaussian noises, c_1, c_2 and c_3 are the constants. The following values for the constants $c_1 = 1$, $c_2 = 12$, $c_3 = 7$ and standard deviation of the noises $\sigma_w = \sigma_v = 2$ are used and the process is initialized at $x_0 = 0$.

A constant velocity model is used for predictions and initial populations of PSO and PSO-AIS are randomly generated form the global best solution. PF is populated with 250 particles and PSO algorithm used 25 particles and 10 PSO iterations. PSO-AIS algorithm is initialized with 20 particles. Five PSO iterations are used for PSO stage of this hybrid algorithm. Later on, these particles act as the initial antibodies of the AIS stage where 5 antibodies each are selected for primary mutation stage and cloning. Cloning operation produced 15 clones from which 5 were selected for secondary mutation. AIS stage of algorithm also runs for 5 iterations. In total, PSO-AIS algorithm uses 250 function evaluations like PF and PSO algorithms. The simulation is run for 100 time steps.

The results in Figure 7.2 show the tracking performance of a generic PF, PSO and PSO-AIS algorithms. The tracking performance of the three algorithms is very similar for the first 50 time steps. An abrupt state jump occurs at the time step $k = 50$. PF lost the track after the state jump as no particles were projected in the proximity of the jumped state. Whereas, in case of PSO, the track is initially lost after the state jump but system was able to recover after an effort of few time steps. In contrast, the tracking performance of PSO-AIS remains good even after the state jump. Estimation errors for the three algorithms are also depicted in Figure 7.2.

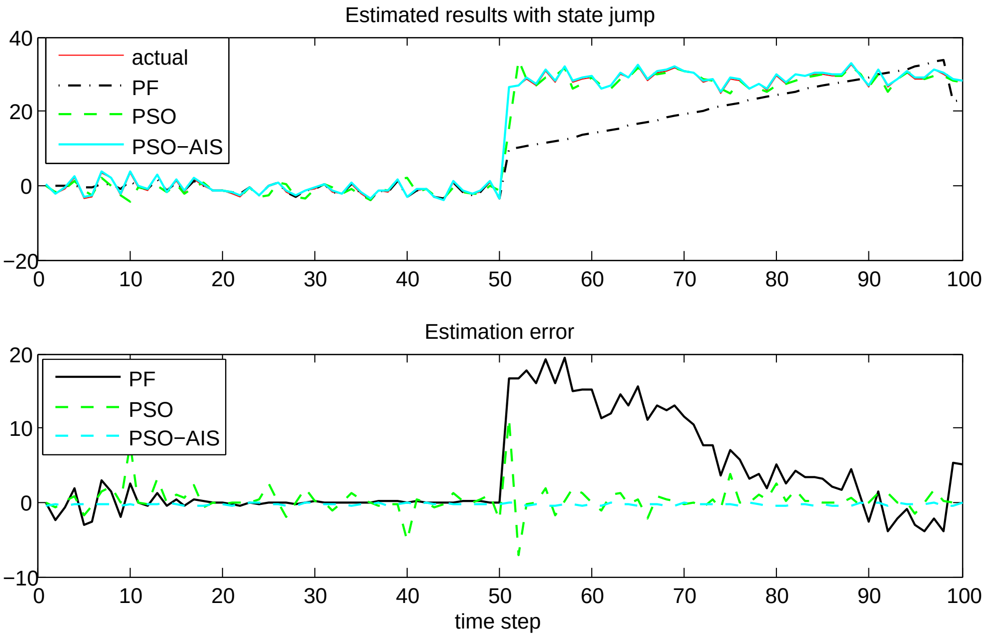


Figure 7.2: Estimation performance with state jump

7.5.2 Dynamic Vehicle System

In this experiment, a *nearly* constant speed turning vehicle is considered [67]. A constant speed turn is a turn with a constant angular speed along a road of constant radius of curvature. The angular rate is allowed to slightly vary. The discrete time model of the dynamic vehicle system is given by the following set of equations:

$$\boldsymbol{x}_k = \begin{bmatrix} 1 & \frac{\sin\omega_{k-1}T}{\omega_{k-1}} & 0 & \frac{1-\cos\omega_{k-1}T}{\omega_{k-1}} & 0 \\ 0 & \cos\omega_{k-1}T & 0 & -\sin\omega_{k-1}T & 0 \\ 0 & \frac{1-\cos\omega_{k-1}T}{\omega_{k-1}} & 1 & \frac{\sin\omega_{k-1}T}{\omega_{k-1}} & 0 \\ 0 & \sin\omega_{k-1}T & 0 & \cos\omega_{k-1}T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{k-1} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \\ 0 & 0 \end{bmatrix} \boldsymbol{w}_k + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \boldsymbol{u}_k \qquad (7.3)$$

$$\boldsymbol{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{v}_k \qquad (7.4)$$

where $\boldsymbol{x}_k = \{x_k \, \dot{x}_k \, y_k \, \dot{y}_k \, \omega_k\}^T$ represents the state vector. x_k and $\dot{x}_k$ represent the x-axis distance and velocity respectively, y_k and $\dot{y}_k$ represent the y-axis distance and velocity respectively, and ω_k is the turn rate of the vehicle at time step k, $\boldsymbol{w}_k \sim N(0, Q)$ and $\boldsymbol{v}_k \sim N(0, R)$ are mutually independent Gaussian noises, $\boldsymbol{u}_k$ is a control input to the turning rate which is considered zero here, T is the discrete time interval when the observations are taken and $\boldsymbol{z}_k$ is a measurement vector. Q and R are the noise covariance matrices with diagonal covariance coefficients of 1^2 and 0.1^2 respectively.

A missing measurement effect is induced at time step $k = 50$ during which vehicle moves faster. The simulation results of a particular run are given in Figure 7.3. Top plot shows the x-axis position and bottom plot gives the y-axis position of the target vehicle produced by the three filtering methods. For PF, the target vehicle is completely lost after measurements become available again, whereas, PSO algorithm has taken some time to recover. In contrast, the PSO-AIS has shown good tracking performance even after this connection loss. Estimation errors are also plotted in Figure 7.4.

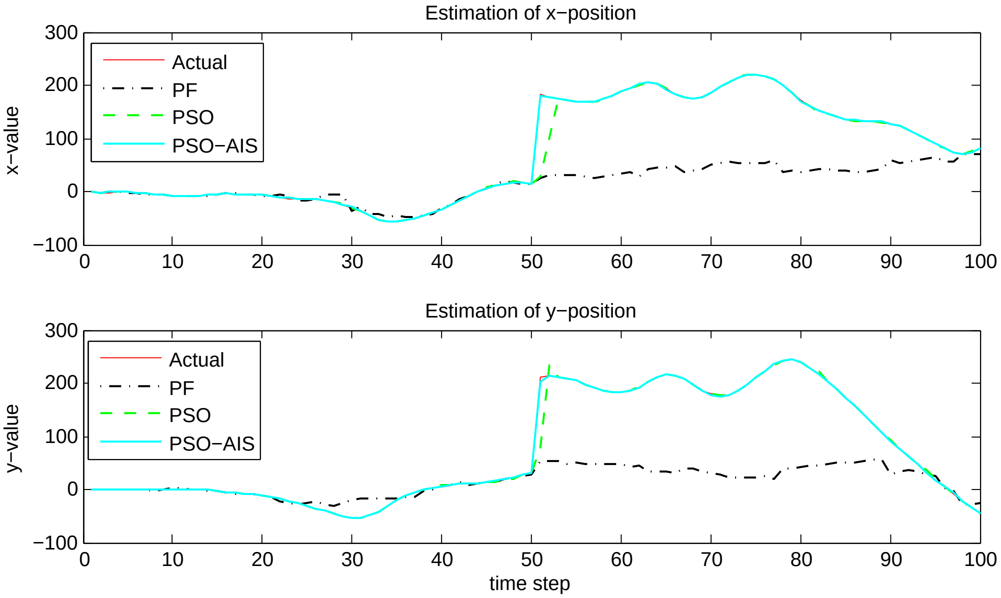


Figure 7.3: Estimated x-axis and y-axis positions

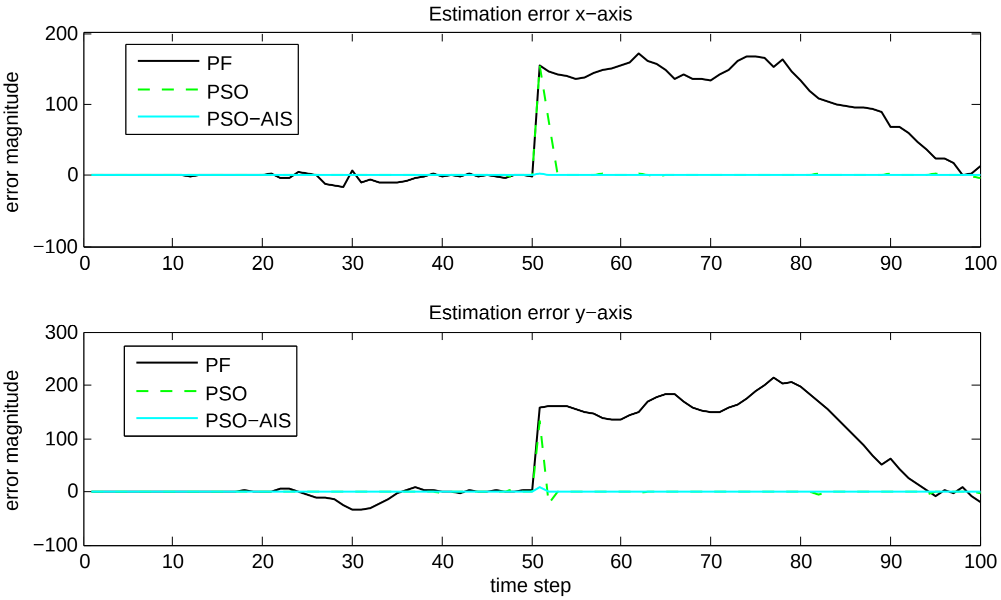


Figure 7.4: Error in estimating x-axis and y-axis position

7.6 Conclusions

In this chapter, a hybrid algorithm based on PSO and AIS is proposed to avoid the degeneracy issue of generic PF. The diversity to search the whole solution space is improved by AIS addition and the PSO improves the initial convergence towards the optimal solution. The algorithm is based on a two stage mutation operation to have a more through exploration of the solution space. This sort of strategy results in improved performance when the target states vary fast or jump abruptly. The simulation results have demonstrated an improved performance of the proposed PSO-AIS algorithm for abruptly varying state estimation problems.

Chapter 8

Performance Analysis of the Proposed Algorithms

In this chapter, a comprehensive performance analysis of the proposed algorithms in terms of scalability, accuracy, precision, recovery and time complexity is investigated. Section 8.1 compares the scalability of the proposed algorithms using a benchmark test function. Section 8.2 describes and compares the quantitative performance of the proposed algorithms for whole body human tracking problem. Section 8.3 elaborates the ability of the proposed algorithms to recover from accumulation of error as well as catastrophic failures. Section 8.4 describes the tests of means and variances to statistically compare the performance of different optimization algorithms considered in this study. Section 8.5 presents the time complexity and run-time complexity of the proposed algorithms. Section 8.6 concludes this chapter with a discussion of the obtained results.

8.1 Preliminary Analysis of the Proposed Algorithms

In this preliminary study, the Rosenbrock banana function, described in Section 5.4, is used. Its global minimum, $f(x^*) = 0$, is located inside a long flat valley which is not trivial to find unless the optimization algorithm has strong local search characteristics. In this way, it resembles to the whole body tracking problem which also requires strong local search ability to refine the pose proposals.

In order to evaluate the scalability of the proposed algorithms, optimization experiments are run with increasing problem dimension. The problem dimension is varied from 2 to

10. An initial population of 20 particles is used and 1000 iterations of optimization runs are carried out. All the optimization algorithms are setup in such a way that about 30,000 function evaluations are performed for each algorithm. The optimization results are listed in Table 8.1 (where ϵ is the Matlab's machine precision error) and plotted in Figure 8.1. From the obtained results it is clear that for all the four algorithms, the optimization performance deteriorates with increasing problem dimension, as the divergence from the optimal value, $f(x^*) = 0$, increases. It is also observed that the optimized function value of the DM-AIS algorithm is closet to the optimal value among the considered algorithms. This is due to the strong local search characteristics of the DM-AIS algorithm.

Table 8.1: Optimization performance of the proposed algorithms

Dim.	Optimization algorithm			
	AP-PSO	MAB	DM-AIS	PSO-AIS
2	ϵ	8.8276×10^{-11}	ϵ	ϵ
3	ϵ	3.0535×10^{-8}	ϵ	ϵ
4	1.7821×10^{-14}	5.5086×10^{-7}	3.3634×10^{-11}	2.8259×10^{-11}
5	2.8037×10^{-7}	1.3917×10^{-6}	1.0216×10^{-7}	9.8721×10^{-8}
6	3.9794×10^{-6}	3.8633×10^{-6}	1.2952×10^{-7}	5.8731×10^{-6}
7	1.8157×10^{-5}	6.0654×10^{-5}	2.5689×10^{-6}	3.2189×10^{-5}
8	3.3444×10^{-4}	1.2237×10^{-4}	7.4609×10^{-6}	1.0059×10^{-4}
9	7.0548×10^{-4}	9.6422×10^{-4}	9.2773×10^{-5}	5.4497×10^{-4}
10	6.3217×10^{-3}	7.3168×10^{-3}	5.6231×10^{-4}	5.8133×10^{-3}

DM-AIS algorithm is further used to investigate the effect of the number of function evaluations on the attained optimal value and is compared with generic PF. A 5-dimensional Rosenbrock banana function is evaluated for different number of function evaluations and results are listed in Table 8.2 and plotted in Figure 8.2. It is clear from the results that the performance of DM-AIS algorithm improves with the increase in number of function evaluations as the divergence from theoretical optimal value decreases and that it outperforms generic PF. A linear curve is also fitted to the DM-AIS performance results which suggests that the performance of DM-AIS algorithm follows an approximately exponential trend with the increasing number of function evaluations. Further, the slope of the curve indicates that this improvement is rapid with increase in function evaluations for DM-AIS algorithm compared to generic PF.

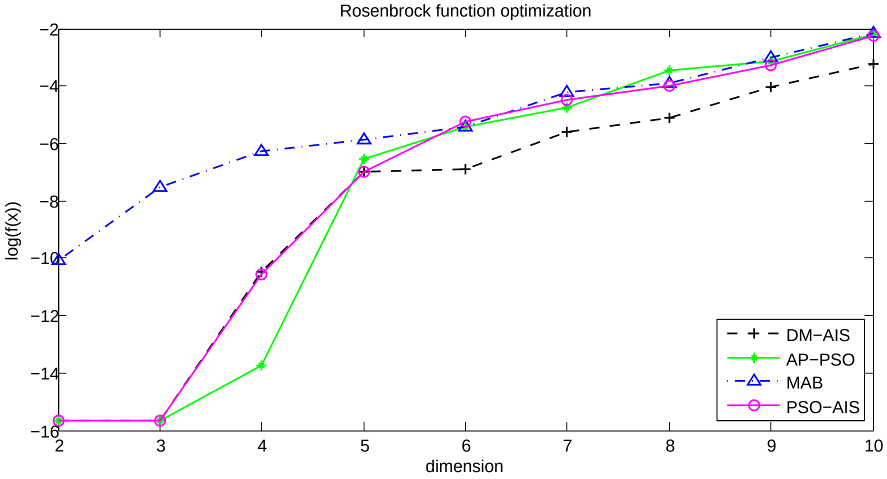


Figure 8.1: Optimization performance of the proposed algorithms for Rosenbrock function evaluation with increasing dimension

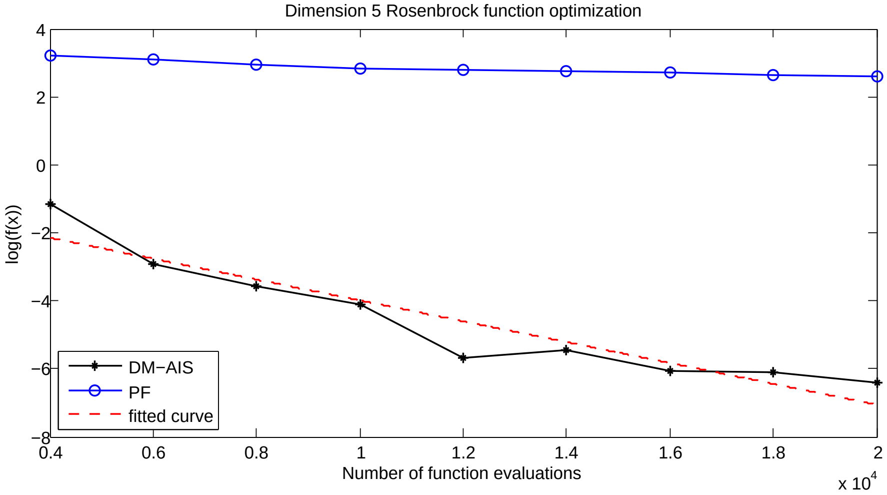


Figure 8.2: Effect of population size on PF and DM-AIS optimization performance

Table 8.2: Optimization performance with increasing function evaluations

Function Evaluations	PF	DM-AIS
4000	1.6213×10^3	7.1733×10^{-2}
6000	1.2529×10^3	1.1632×10^{-3}
8000	900.4714	2.799×10^{-4}
10000	650.1888	7.7257×10^{-5}
12000	603.2917	2.1139×10^{-6}
14000	560.0270	3.4507×10^{-6}
16000	510.3718	8.6309×10^{-7}
18000	454.6889	8.1311×10^{-7}
20000	382.1749	4.0487×10^{-7}

8.2 Quantitative Performance Analysis: Whole Body Human Tracking

In these experiments, Lee Walk and HumanEva II data sets are used, as they are more frequently reported data sets in AHT research literature. Lee Walk, like HumanEva I, shows the imperfect ground truth data due to software synchronization of motion capture and video capture systems (evident by results in Figure 8.4). HumanEva II data set is used because of its accuracy of motion capture data, as the data set is acquired using a latest Vicon MX System with twelve 1.3 M-pixel cameras. Moreover, hardware synchronization of video and motion capture systems is used, which results in better temporal scaling and offset.

In HumanEva data set, Lee Walk sequence, where a person walking in a circle in front of a multi-camera setup, is considered [132]. A total of 150 frames are analyzed. In the first frame, the human model is initialized using ground truth motion capture data. The sizes of the body segments are predefined. Statistical background subtraction with a Gaussian mixture model is used to extract the object of interest. The Brown University computational framework [132] is used to compare the performance of various tracking algorithms: PF, APF, PSO, AP-PSO, MAB, DM-AIS and PSO-AIS. The optimized parameters for each algorithm are utilized.

In another set of experiments, two video sequences, S_2 and S_4, from HumanEva-II data set are used to evaluate the performance of the proposed algorithms. The human model used in this case has 40-states. It augmented the model described in Section 3.3 with one additional degree of freedom for each of the elbow, knee and torso joints and 1-degree of freedom for the added wrist and ankle joints, that makes a total addition of 9-states.

The predictions for the next frame ($\boldsymbol{x}_{i-}^{t}$) at time t are made using a temporal model. A zero velocity motion model $\boldsymbol{x}_{i-}^{t} = \boldsymbol{x}_{i}^{t-1} + \epsilon$ is used to predict the joint angles, states of the system, at time t. The noise ϵ is drawn form a Gaussian distribution whose standard deviation is equal to the maximum inter-frame joint angle difference. It accounts for model uncertainties and produces a random velocity effect. The biomechanical constraints are used as hard priors to constrain the search space and avoid implausible human poses. Any particle exceeding the joint angle limits is discarded and randomly initialized for the next frame/iteration.

The 3D error between the predicted and the ground truth human pose is computed following the procedure in [12]. Virtual markers ($m = 15 \in M$) are placed on the joints of the subject. For each particle x_{i}^{t}, the human pose error $e_{human}(\boldsymbol{x}_{i}^{t}, \tau^{t})$ is computed as the average distance between the virtual markers placed on the predicted human pose $\boldsymbol{x}_{i}^{t}$, and those placed on the true pose τ^{t} as:

$$e_{human}(\boldsymbol{x}_{i}^{t}, \tau^{t}) = \frac{\sum_{m \in M} \| m(\boldsymbol{x}_{i}^{t}) - m(\tau^{t}) \|}{|M|} \tag{8.1}$$

where $m(\boldsymbol{x}_{i}^{t})$ is the 3D location of marker m for the predicted pose and $m(\tau^{t})$ is the location of the marker on the true pose.

The number of fitness evaluations was fixed to 2500 for all the considered algorithms to achieve a fair evaluation of algorithm performance. To maintain this constant number of function evaluations, PF was initialized with 2500-particles, APF was initialized with 500-particles and 5-annealing layers were used. AP-PSO was initialized with 500-particles and 5-iterations of PSO run were used.

Similarly, for MAB algorithm, 300 particles with 5 iterations were used. It was observed that a further $50 - 60\,\%$ of the original particles were selected for the neighborhood search. This makes the total number of function evaluations approximately equal to that of the other considered algorithms.

For DM-AIS algorithm, 250 antibodies make the initial population and 50 antibodies each were selected for primary mutation and cloning. Cloning setup generates 300 clones of which 50 were selected for the secondary mutation. DM-AIS algorithm was run for 5 optimization cycles for each video frame. This way the whole DM-AIS setup was also resulted in 2500 function evaluations per frame.

PSO-AIS algorithm was initialized with 200 particles and 5 iterations were used for PSO stage of hybridization. These particles were then used as the initial antibodies for the

AIS stage where 50 antibodies were selected for cloning and primary mutation. A total of 150 clones were produced from them 50 were selected for secondary mutation stage. This way the whole AIS setup resulted in 300 function evaluations for each iteration. A total of 5 AIS iterations were considered to make equal likelihood evaluations, 2500 function evaluations, across all the considered algorithms.

The tracking performance in terms of mean and standard deviation is also compared with Partitioned Sampled Annealed Particle Filter (PSAPF) [13] and HPSO [61] results. Both these methods do a hierarchical decomposition of the state space and then sequentially estimate body parts and eventually the whole pose. PSAPF uses APF for hierarchial search while HPSO uses PSO for this hierarchical pose estimation.

The Most Likely/Appropriate Pose (MAP) error is evaluated for each frame using Equation 8.1 to quantitatively compare the tracking performance in terms of mean and standard deviation of error. The errors at each frame are combined to compute the overall error of the sequence as:

$$E_{Overall} = \frac{1}{K}\sum_{k=1}^{K} e_k(x_i^k, \tau^k) \tag{8.2}$$

where K is the total number of frames used in an experiment.

All these experiments are highly stochastic in nature and produce different results for different runs for the same configuration of parameters. So, each experiment is run 5 times for each of the video sequences to get a measure of the performance consistency and repeatability and the resultant mean error along with the standard deviations are listed in Table 8.3. The table shows the mean value of MAP error over the 150 frames for Lee Walk (Sequence S_5). For HumanEva II subjects, S_2 and S_4, 200 frames are considered for analysis. The results indicate that DM-AIS algorithm has performed the best in terms of mean error and standard deviation measures among the considered tracking algorithms for all the considered sequences.

Figure 8.3, shows the tracking performance of the considered algorithms for a specific run in terms of MAP error for Lee Walk (Sequence S_5). It can be seen that the performance of PF is inferior to the other considered algorithms. In fact, the error builds up in this run and tracking progressively diverges. This is due to the lack of a local search mechanism in PF framework. It can also be noticed that the errors of first few frames are small, the reason for this is the use of ground truth data for initialization in the first frame. The MAP error scores suggest that the proposed algorithms performed better than the other considered algorithms. DM-AIS has produced the best performance which can be

Table 8.3: Comparison of MAP error/frame (in mm)

Algorithm	Function Eval.	S_2	S_4	S_5
PF	2500	93.69 ± 13.81	86.6 ± 7.11	57.144 ± 19.639
APF	2500	71.14 ± 5.39	60.15 ± 3.13	43.433 ± 9.284
PSAPF [61]	7200	-	-	48.1 ± 12.8
HPSO [61]	7200	-	-	46.5 ± 8.48
AP-PSO	2500	69.64 ± 6.21	58.83 ± 3.88	39.87 ± 7.201
MAB	2500	70.25 ± 7.03	58.91 ± 5.17	40.87 ± 8.001
DM-AIS	2500	67.85 ± 6.01	56.64 ± 3.61	37.81 ± 4.883
PSO-AIS	2500	69.13 ± 6.83	58.19 ± 4.89	38.97 ± 6.239

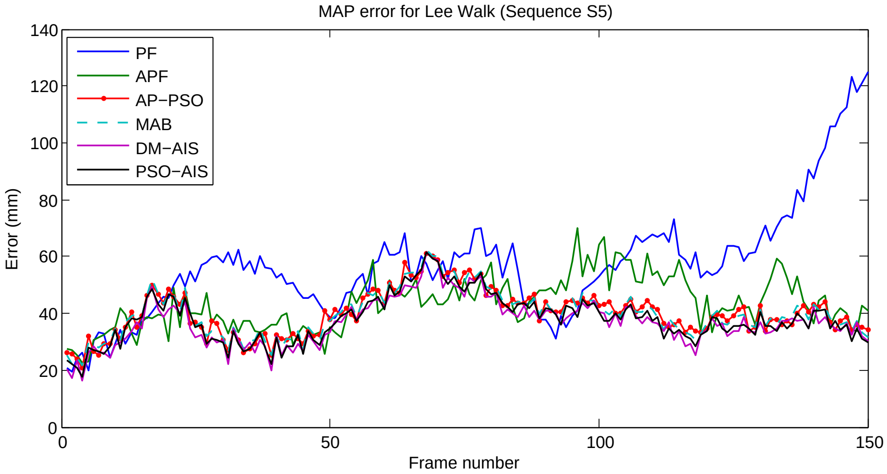


Figure 8.3: Comparison of MAP error of various algorithms for Lee Walk Sequence S_5

attributed to its strong local search ability due to cloning and further enhanced local and global search abilities imparted by primary and secondary adaptive differential mutation operation.

Occasional spikes in error are noticed in frame-to-frame error plots in Figure 8.3. Self-occlusion is the driving reason for these fluctuations in measurement error. There are instances in the video sequence where body parts are overlapping each other in such a way that it is difficult to distinguish them and are hence erroneously detected. In this situation it is difficult for the image likelihood models, silhouette and edge likelihoods, used in this study to identify the limbs. This occlusion occurs frequently in upper body tracking as the arms and torso overlap. The situation is more detrimental for silhouette likelihood which effectively does not have the capability to model the internal structure.

The tracking images for selected frames for a run of PF, APF, PSO and AP-PSO are shown in Figure 8.4. Here again, the PSO algorithm is initialized with 500 particles and 5 iterations were used resulting in 2500 function evaluations. The optimized parameters for PSO were used. In this figure, the black cylindrical model represents ground truth pose and the colored cylindrical model represents the estimated pose. It is noticed that some of the error is due to the poor image observations and incorrect ground truth data. Such as, in Frame# 60, it is hard to distinguish the left leg from the background. A close examination indicates that the estimated leg position is closer than the provided ground truth leg position. This situation is also obvious in Frame# 75, where estimated left leg and left arm are more coinciding with the image observations than the ground truth pose. Similarly, in the last frame, Frame#150, the head is erroneously marked by the ground truth data. It is also observed that APF, PSO and AP-PSO were able to recover after the loss of tracking as can be seen by comparing results for Frame# 90 & 105.

The mean and standard deviation of the error for sequences S_2 and S_4 are also listed in Tables 8.3. The comparative frame-to-frame mean error tracking performance is also plotted in Figure 8.5 and Figure 8.6 for HumanEva II sequences S_2 and S_4 respectively. The mean and standard deviation results show improved performance of the proposed algorithms over generic PF. The performance difference between the proposed algorithms and APF is not obvious, since the difference between their overall errors are within the standard deviation. A cyclic variation of error is observed in the results. The reason for this is self-occlusion, which occurs frequently due to articulated nature of motion of the human limbs. In this situation employed likelihood functions are not able to model the internal structure.

The required number of particles to search the whole solution space grows exponentially for PF algorithm [87]. In case of the proposed algorithms, which are based on exploration and exploitation, the requirement for the number of particles is reduced. This is one of the reasons for the good tracking performance of the proposed algorithms. Some of the tracking frames for AP-PSO tracking are shown in Figure 8.7 and 8.8 for sequences S_2 and S_4 respectively.

It is observed that more error occurs in upper body tracking than the lower limbs. The reason for this could be the higher rate of motion of upper body compared to the load bearing lower limbs. A more robust and computationally efficient image likelihood model which is less susceptible to background and lighting conditions is likely to improve the performance of the tracking framework. Moreover, in this study cylinders are used to represent the human limbs. It is reasonable to expect that a more realistic body model

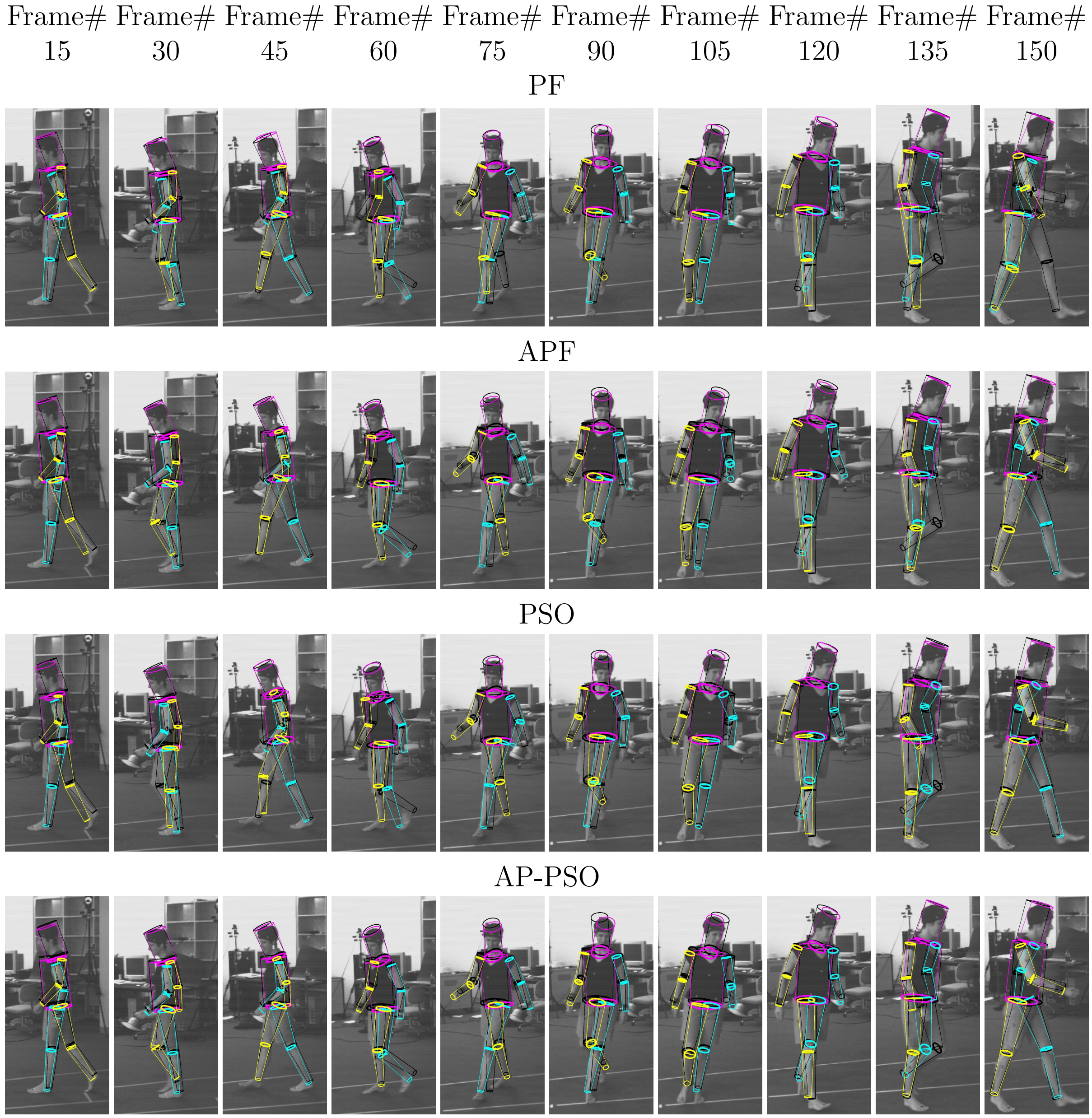


Figure 8.4: Comparison of PF (1^{st} row), APF (2^{nd} row), PSO (3^{rd} row) and AP-PSO (4^{th} row) tracking performance

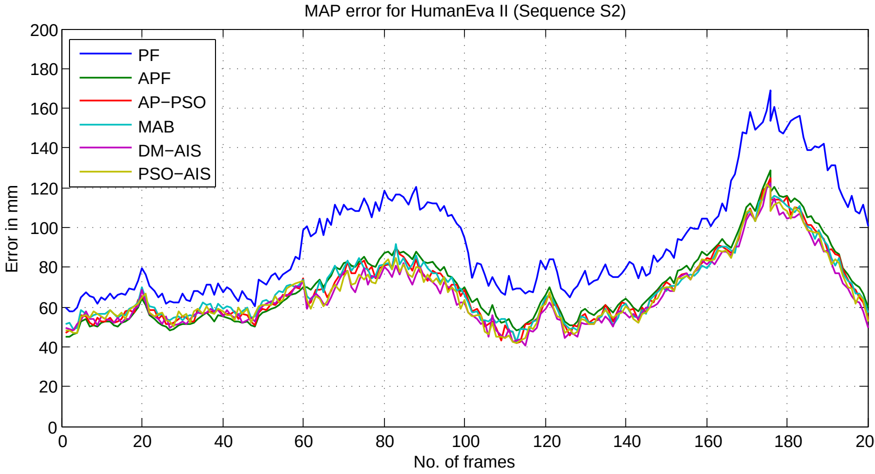


Figure 8.5: Comparison of MAP error of various algorithms for HumanEva II Sequence S_2

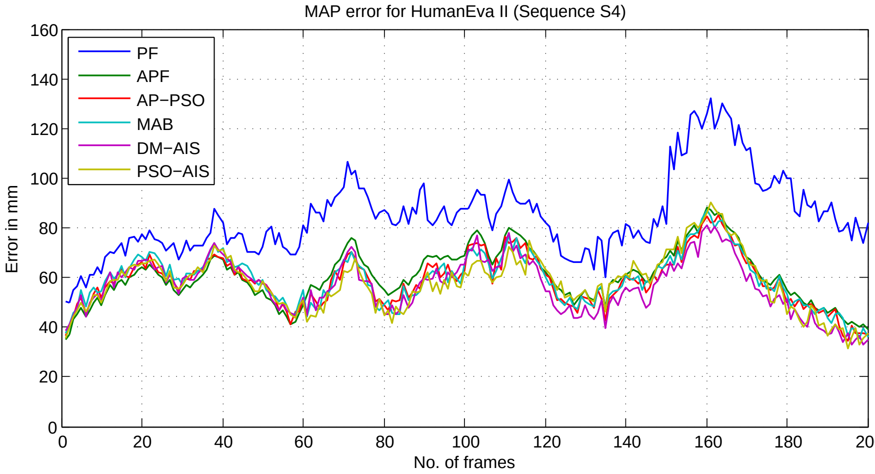
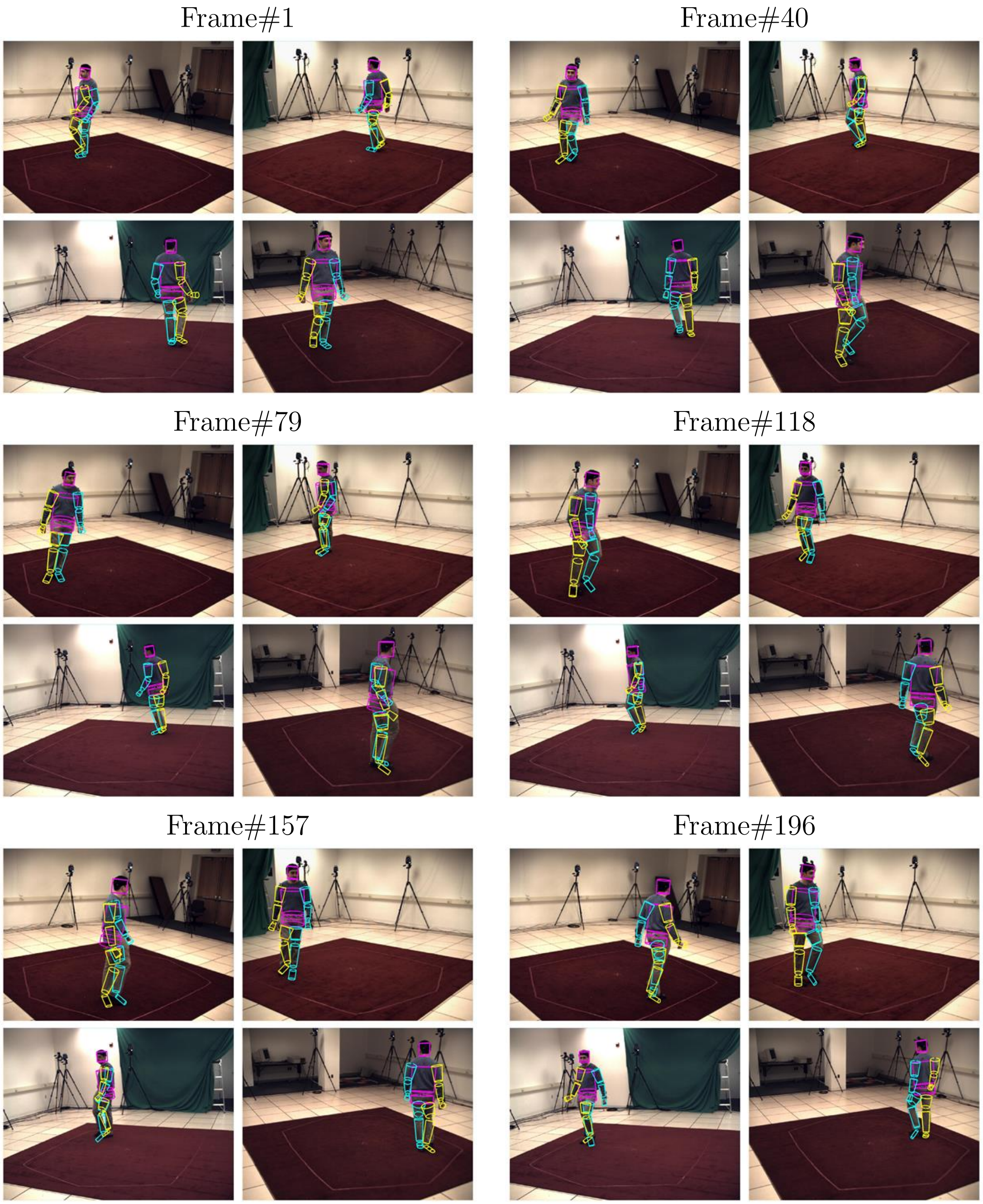


Figure 8.6: Comparison of MAP error of various algorithms for HumanEva II Sequence S_4

can result in better tracking performance.

Although the video sequences used in this study are generally free of occlusions as they are captured in laboratory settings but strong illumination poses some challenges, such as shadows can confuse the background subtraction algorithm. Edge likelihood measures may suffer due to these illumination effects and may result in spurious edges which can deteriorate the tracking performance.



Frame#1

Frame#40

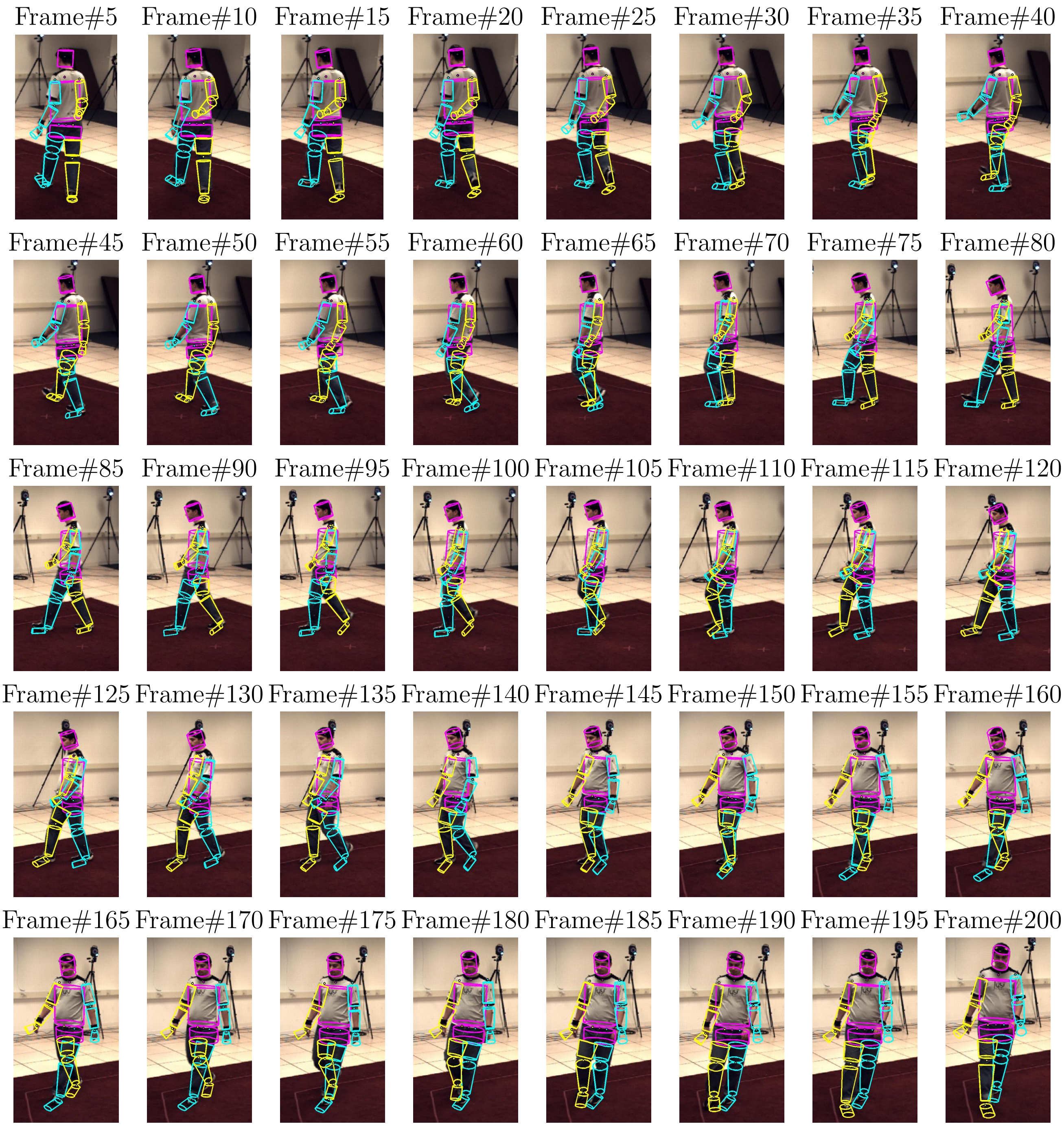
Frame#79

Frame#118

Frame#157

Frame#196

Figure 8.7: Tracking of S_2 using AP-PSO with 500 particles and 5 PSO iterations



Frame#5 Frame#10 Frame#15 Frame#20 Frame#25 Frame#30 Frame#35 Frame#40

Frame#45 Frame#50 Frame#55 Frame#60 Frame#65 Frame#70 Frame#75 Frame#80

Frame#85 Frame#90 Frame#95 Frame#100 Frame#105 Frame#110 Frame#115 Frame#120

Frame#125 Frame#130 Frame#135 Frame#140 Frame#145 Frame#150 Frame#155 Frame#160

Frame#165 Frame#170 Frame#175 Frame#180 Frame#185 Frame#190 Frame#195 Frame#200

Figure 8.8: Tracking results of S_4 using AP-PSO with 500 particles and 5 PSO iterations

8.2.1 Effect of Population Size on Tracking Performance

Experiments are also run with different numbers of AP-PSO and MAB populations. The MAP error results are listed in Table 8.4 for an average of 5-runs for all 150 frames. Table 8.4 shows that the mean and standard deviation of MAP error decreases with the increase in population. Further, the MAP error is decreasing almost linearly as depicted in Figure 8.9. The frame-to-frame MAP error for a specific run is also shown in Figure 8.10 and 8.11 for AP-PSO and MAB algorithms respectively. It confirms that the error remains bounded as the number of particles decrease and the amount of deviations from the ground truth values decrease *on average* as the number of particles increase.

Table 8.4: Comparison of different AP-PSO and MAB configurations

Population	MAP error/frame (in mm)	
	AP-PSO	MAB
100	43.547 ± 9.327	42.743 ± 9.567
300	41.31 ± 8.352	40.87 ± 8.001
500	39.87 ± 7.201	39.14 ± 7.826

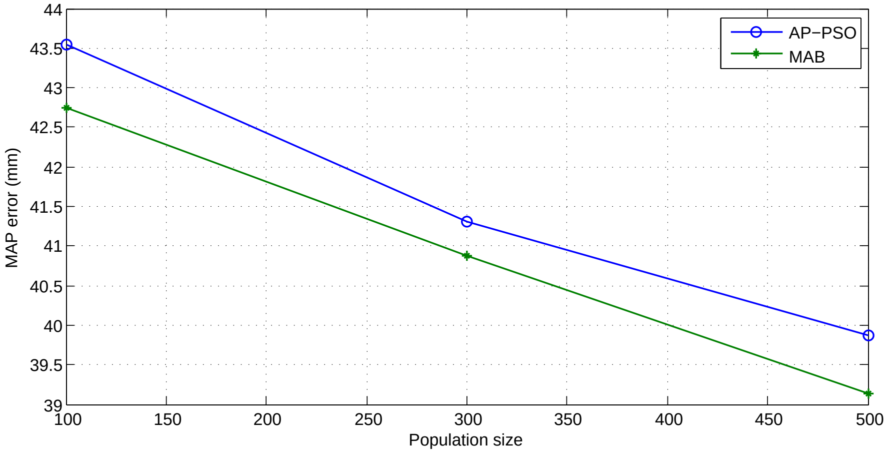


Figure 8.9: Change in MAP error with increase in population size

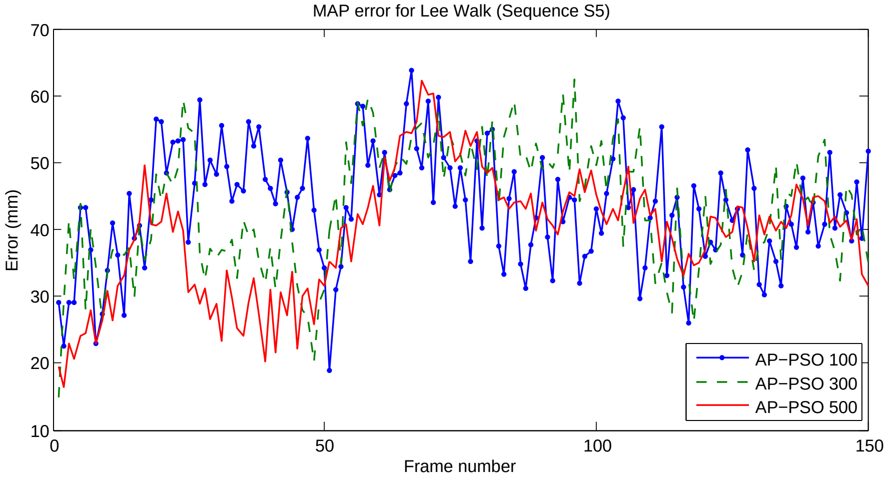


Figure 8.10: Comparison of MAP error for three AP-PSO population sizes

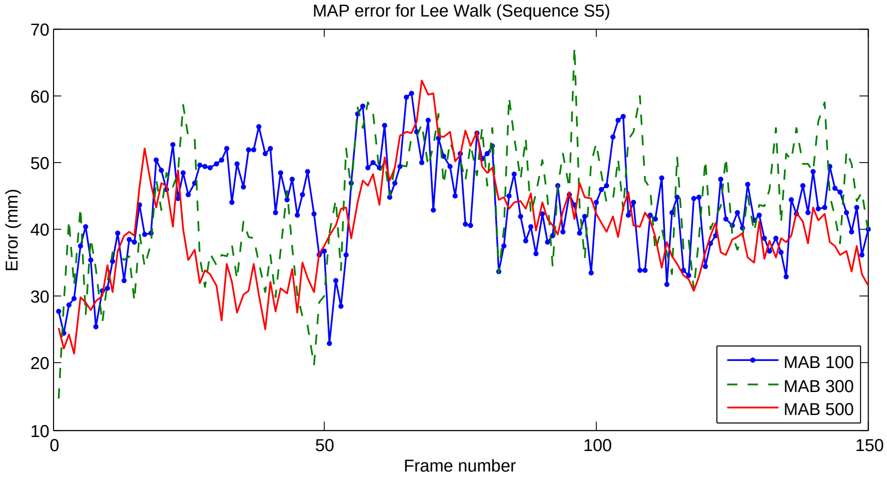


Figure 8.11: Comparison of MAP error for three MAB population sizes

8.3 Recovery Characteristic of the Proposed Algorithms

The performance of the proposed algorithms is also tested to evaluate their recovery characteristics. This attribute is useful in situations where a sudden change in motion type, a loss of image data or other catastrophic or algorithmic failures, which results in a growth of MAP error, occurs. It may even help to reduce the effect of the accumulation of error which results in progressive deterioration of tracking performance and eventual failure.

A long sequence from HumanEva II dataset (Subject S_2), consisting of 1200 frames, is considered in this experiment. The sequence consists of three different motion types: walking, jogging and balancing. Each of these activities require a distinct action model for optimal tracking performance. Instead, a general action model is considered throughout this experiment. In the beginning of the sequence, the subject is walking in a circle resulting in frequent overlapping of body parts which may not be observable by any of the 4-cameras used in observation and results in self-occlusion. At Frame#375, the subject starts to jog in the same circular path. The jogging phase continues till Frame#750, where the subject starts to balance himself on one leg alternating between the left and right legs and this continues till the end of the sequence.

In this experiment, the number of likelihood evaluations are kept constant to obtain a fair comparison of performance across all the proposed optimization algorithms. Initial population and number of optimization iterations for PF, APF, AP-PSO, MAB, DM-AIS and PSO-AIS are selected in such a manner that each results in 2500 likelihood evaluations per frame. Due to the stochastic nature of the experiment, it was repeated five times to get a measure of performance accuracy and repeatability. The results are listed in Table 8.5 in terms of mean tracking error and corresponding standard deviation per frame for each motion type and for the overall sequence for each of the considered set of optimization algorithms. A plot of the frame-to-frame mean error measure for each algorithm is illustrated in Figure 8.12.

The mean tracking error performance of PF is significantly worse than the other algorithms. Initially, PF tracking performance deteriorates progressively with increasing number of frames due to accumulation of error during walking phase. A change in motion pattern, from walking to jogging, results in abrupt deterioration of PF tracking performance. This divergence continues to effect its performance during the balancing phase as well. PF fails to recover from both progressive and catastrophic error growth. This lack of recovery, due to error accumulation or catastrophic failure, can be linked to the 'degeneracy' challenge of PF which results in loss of the diversity of solution space search.

Table 8.5: MAP error comparison for HumanEva II Sequence S_2

Algorithm	MAP error/frame (in mm)			
	Walk	Jog	Balance	Overall
PF	102.03 ± 13.93	177.58 ± 27.52	243.96 ± 45.93	181.49 ± 29.67
APF	75.80 ± 5.46	85.44 ± 5.91	86.22 ± 5.61	82.72 ± 5.57
AP-PSO	68.16 ± 6.33	78.86 ± 6.81	76.65 ± 6.19	74.69 ± 6.42
MAB	70.92 ± 6.89	80.28 ± 7.55	78.99 ± 6.29	76.87 ± 6.74
DM-AIS	65.11 ± 5.51	74.66 ± 5.47	72.24 ± 5.11	70.77 ± 5.26
PSO-AIS	66.55 ± 5.67	77.27 ± 5.33	75.06 ± 5.29	73.09 ± 5.42

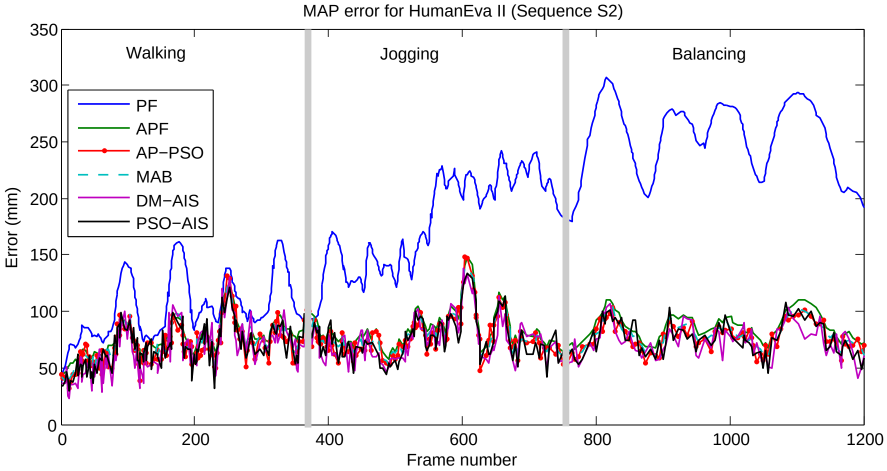


Figure 8.12: MAP error comparison of various algorithms for a multi-activity sequence

On the other hand, among the considered algorithms, DM-AIS results in the least mean error and standard deviation due to its good local search ability for all motion types. The performance of the proposed algorithms and APF also deteriorates during the jogging phase but to a lesser extent compared to PF. Further, it is noted that the performance is improved during the balancing phase again which depicts recovery characteristics of the proposed algorithms. The improved performance is a result of less self-occlusions during the balancing phase due to the motion type and positioning of the subject with respect to the four used cameras. It also indicates that the proposed optimization algorithms are insensitive to accumulation of error.

8.4 Statistical Analysis of Performance

A formal analysis of accuracy and reliability of the proposed algorithms in comparison to four published algorithms [12, 36, 61, 129] is performed using hypothesis testing techniques [93]. *ANOVA* and *t-test* are performed to evaluate the accuracy of the proposed algorithms and *F-test* is performed to compare the reliability of the proposed algorithms [93]. The data, means μ and standard deviations σ, for these statistical comparisons are obtained from Table 8.3 for subject S_5.

The null hypothesis considered for *t-test* asserts that the two means are equal. A right-handed *t-test* is performed with a significance level of $\alpha = 0.05$ which gives a rejection threshold of 1.86. The t-statistics of the pairwise comparison of the means is listed in Table 8.6.

Table 8.6: Pairwise multiple *t-test* statistics

Algorithm	PF	APF	PSAPF	HPSO	AP-PSO	MAB	DM-AIS	PSO-AIS
PF	-	1.4114	0.8627	1.1126	1.8466	1.716	2.1363	1.9721
APF	-	-	0.66	0.5454	0.6781	0.4676	1.1986	0.8922
PSAPF	-	-	-	0.233	1.253	1.071	1.6795	1.4337
HPSO	-	-	-	-	1.3326	1.0798	1.9858	1.5993
AP-PSO	-	-	-	-	-	0.2077	0.5294	0.2112
MAB	-	-	-	-	-	-	0.73	0.4187
DM-AIS	-	-	-	-	-	-	-	0.3353
PSO-AIS	-	-	-	-	-	-	-	-

The comparison of mean of PF with the means of the proposed algorithms provide *sufficient* evidence, at $\alpha = 0.05$, to conclude that the means are different. A *t-test* statistic of zero indicates the means are identical. Any departure from zero indicates that means differ, with a larger value indicating a greater spread. With the relaxation in level of significance, it can be concluded that the means of the proposed algorithms significantly differ from the means of PF, PSAPF and HPSO.

Analysis of variance (*ANOVA*) test is also performed to compare the difference between group means. *ANOVA* statistic is a ratio of variability *among/between* the sample means and variability *around/within* the distribution. If the variability *between* the sample means (distance between overall mean) is relatively large compared to the variance *within* the samples (internal spread), the ratio will be much larger than 1. This indicates that the samples most likely did *not* come from a common population; thereby rejecting the *null hypothesis* which states that the means are equal.

For *ANONA* comparison, five different groups are formed. The first four groups contain one of the published algorithm [12, 36, 61, 129] and the 3-best performing (AP-PSO, DM-AIS and PSO-AIS) of the proposed algorithms. The last group contains all four proposed algorithms. *ANOVA* comparison results are given in Table 8.7.

Table 8.7: *ANOVA* comparison statistics

Algorithm	*ANOVA* Statistic
PF+3-best	3.3609
APF+3-best	0.5862
PSAPF+3-best	1.5763
HPSO+3-best	1.6313
MAB+3-best	0.1901

ANOVA comparison among PF and 3-best proposed algorithms provides *sufficient* evidence at $\alpha = 0.05$, rejection region threshold is at 3.24, to conclude that at least two algorithm means differ. *ANOVA* result of the comparison of the four proposed algorithms gives a smaller score suggesting that the means are similar. On these basis, it can be concluded that the difference between algorithm means of *ANOVA* comparison among PF and 3-best proposed algorithms is the result of PF mean.

Similarly, if we increase the level of significance, $\alpha = 0.25$ which results in a rejection threshold of 1.5, it can be concluded that the means of the other considered algorithms also differ from the proposed algorithms except APF.

F-test is used for the analysis of variance. Again, the null hypothesis asserts equal variances. A level of significance $\alpha = 0.05$ gives a rejection threshold of 6.39. The *F-test* statistics are listed in Table 8.8.

Comparison of variance of PF with proposed algorithms provides *sufficient* evidence, at $\alpha = 0.05$, that the proposed algorithms have smaller variances which implies to their higher reliability or robustness. It is further noted that an *F-test* statistic of one indicates that the variances are similar. A value larger than one indicates a mismatch of variances with larger value indicating more difference. Relaxing the level of significance to $\alpha = 0.25$ results in a rejection threshold of 2.06. It indicates that variances of DM-AIS and PSO-AIS are smaller than APF. Similarly, variance of DM-AIS is smaller than HPSO and variances of all the proposed algorithms are smaller than PSAPF.

Table 8.8: Pairwise multiple *F-test* statistics

Algorithm	PF	APF	PSAPF	HPSO	AP-PSO	MAB	DM-AIS	PSO-AIS
PF	-	4.4747	2.3541	5.3635	7.4379	6.0249	16.1758	9.9085
APF	-	-	1.9009	1.1986	1.6622	1.3464	3.6149	2.2143
PSAPF	-	-	-	2.2784	3.1596	2.5594	6.8714	4.2091
HPSO	-	-	-	-	1.3868	1.1233	3.0159	1.8474
AP-PSO	-	-	-	-	-	1.2345	2.1748	1.3322
MAB	-	-	-	-	-	-	2.6848	1.6446
DM-AIS	-	-	-	-	-	-	-	1.5098
PSO-AIS	-	-	-	-	-	-	-	-

8.5 Time Complexity of the Proposed Algorithms

A formal time complexity analysis of the proposed algorithms is carried out in this section. Time complexity is a description of the asymptotic behavior of running time as input size tends to infinity. It gives an upper limit on the growth of a function as the system size increases. It does not depend on the implementation details, such as programming language, quality of compiler or speed of the computer.

In our analysis, n is the number of particles and m is the number of iterations used by the optimization algorithm. For PSO-AIS algorithm, we consider $m = m_1 + m_2$; where m_1 is the number of iterations for the PSO and m_2 is the number of iterations for the AIS parts of the hybrid algorithm. The number of operations used to find fitness value of a single particle is represented by α and it depends on the cost function. The results of time complexity analysis are listed in Table 8.9.

The rows marked ***Init.*** in Table 8.9 give the number of operations required for initialization, the rows marked ***Loop*** give the number of operations required inside the loop for a single iteration and the rows marked ***Total*** give the total number of operations used by the optimization algorithm. The last column gives the asymptotic time complexity of the algorithm. The analysis suggests that the complexity of all the proposed optimization algorithms is on the order of the product of the input size and the number of optimization iterations.

Run-time complexity of the proposed algorithms is also measured using a five dimensional Rosenbrock function. Both the number of inputs and the number of optimization iterations are varied and the results are depicted in Table 8.10. The results are in accordance with the theoretical findings of Table 8.9, that is, the execution time approximately

Table 8.9: Time complexity of the proposed algorithms

Algorithm	Step	Operations	Complexity
AP-PSO	Init. Loop Total	$2 + 3n + \alpha \cdot n$ $29n + \alpha \cdot n + 1$ $(2 + 3n + \alpha \cdot n) + m(29n + \alpha \cdot n + 1) + 1$	$O(m \times n)$
MAB	Init. Loop Total	$4n + \alpha \cdot n$ $23n + 2\alpha \cdot n + 1$ $(4n + \alpha \cdot n) + m(23n + 2\alpha \cdot n + 1) + 1$	$O(m \times n)$
DM-AIS	Init. Loop Total	$2n + \alpha \cdot n$ $13n + 3\alpha \cdot n + 1$ $(2n + \alpha \cdot n) + m(13n + 3\alpha \cdot n + 1) + 1$	$O(m \times n)$
PSO-AIS	Init. Loop Total	$2 + 3n + \alpha \cdot n$ $(29n + \alpha \cdot n + 1) + (13n + 3\alpha \cdot n + 1)$ $(2+3n+\alpha \cdot n)+m_1(29n+\alpha \cdot n+1)+m_2(13n+3\alpha \cdot n + 1) + 1$	$O(m \times n)$

doubles as the number of inputs double. It also doubles as the number of optimization iterations become twofold.

Table 8.10: Run-time complexity of the proposed algorithms (input x iterations)

Algorithm	Execution time in seconds				
	20 × 500	40 × 500	40 × 1000	80 × 2000	160 × 4000
AP-PSO	0.3357	0.6639	1.2835	5.2159	20.9936
MAB	0.4329	0.8581	1.6905	6.6718	27.0891
DM-AIS	0.3491	0.6877	1.3686	5.4401	21.6734
PSO-AIS	0.3473	0.6911	1.3615	5.4284	21.6755

Run-time of the proposed algorithms are also computed with increasing problem dimension. An average run-time of 25 executions of each algorithm for each dimension is recorded in Table 8.11. Linear and exponential curves are fitted to these run-time results and are shown in Figure 8.13. The resultant coefficients are tabulated in Table 8.12. The coefficients b and A represent the initial cost incurred and the coefficients a and λ represent the run-time complexity. The results indicate that run-time increases with problem dimension. It can be seen that AP-PSO is least run-time complex followed by MAB which is followed by PSO-AIS. DM-AIS is the most run-time complex among the proposed algorithms as problem dimension increases.

Table 8.11: Run-Time of the proposed algorithms with increasing problem dimension

Algorithm	Execution time in seconds with problem dimension							
	5	10	15	20	25	30	35	40
AP-PSO	0.3388	0.3491	0.3504	0.3506	0.3533	0.3591	0.3630	0.3741
MAB	0.4277	0.4371	0.4319	0.4361	0.4431	0.4425	0.4626	0.4744
DM-AIS	0.3529	0.3634	0.3561	0.3778	0.3654	0.3913	0.3993	0.4009
PSO-AIS	0.3454	0.3510	0.3521	0.3564	0.3602	0.3741	0.3826	0.3862

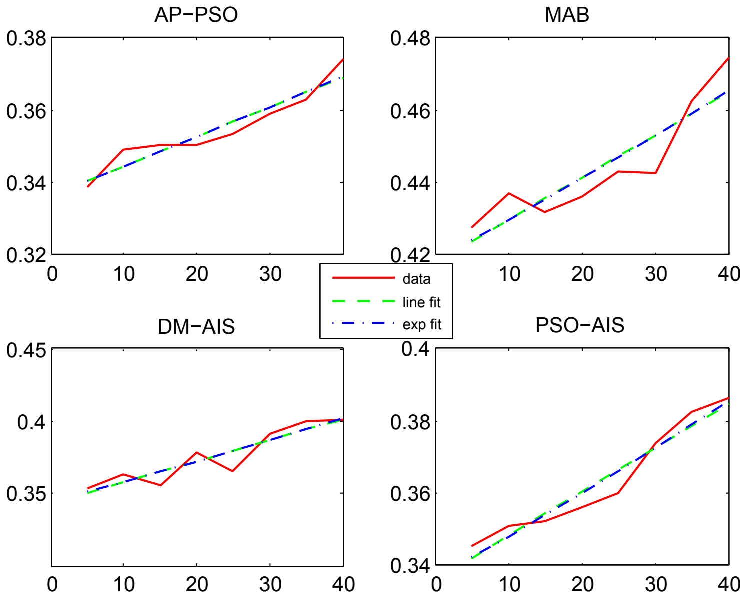


Figure 8.13: Linear and exponential curve fitted to increasing dimension run-time

8.5.1 Time Complexity for AHT

The current Matlab implementation of the proposed optimization algorithms with 500 particles and 5 iterations, a total of 2500 function evaluations, takes about 65 seconds per frame on a 2.2 GHz Core 2 Duo CPU equipped with 4 GB of RAM for Lee Walk sequence (S_5).

Tracking framework uses silhouettes and edges as the observations. There is no simple detector that can give silhouette and edges directly from the image observations. Therefore, low level image processing is required to collect these image features which is computationally expensive. In generative approaches, the time taken by the tracking algorithm

Table 8.12: Resultant coefficients of line and exponential fit

Algorithm	Line Fit $f(x) = ax + b$		Exponential Fit $f(x) = Ae^{\lambda x}$	
	a	b	A	λ
AP-PSO	0.0008	0.3365	0.3366	0.0023
MAB	0.0012	0.4180	0.4183	0.0027
DM-AIS	0.0014	0.3433	0.3442	0.0039
PSO-AIS	0.0012	0.3360	0.3366	0.0034

(proposed metaheuristic optimization algorithms in our case) is dominated by the number of likelihood evaluations. In our simulations, this computational cost is incurred by the 2500 likelihood evaluations per frame for each of the four cameras in use. Each of the 2500 particles used, generate a 3D proposal pose that needs to be projected onto the image plane in the form of cylinders. From these projected poses the silhouette and edge statistics are collected for likelihood computation.

As far as HumanEva II data set sequences (S_2 and S_4) are concerned, it takes approximately 83 seconds per frame. This increase in time is due to the projection of additional cylinders to represent hands, feet and another cylinder to represent an additional degree of motion for the torso. In general, time taken by the optimization algorithms is almost insignificant as compared to the total time taken by the tracker to process one video frame.

Time taken by PF and APF algorithms is 204 seconds per frame when 7200 function evaluations are used [61]. Similarly, time taken by PSAPF is 120 seconds and time taken by HPSO to process one frame of Lee Walk sequence is 76.8 seconds where both algorithms use 7200 function evaluations [61]. Both PSAPF and HPSO algorithms execute faster per function evaluation but more function evaluations are required to achieve the same level of accuracy as the proposed algorithms. A summary of execution times for the Lee Walk sequence S_5 is given in Table 8.13.

8.6 Conclusions

In this chapter, performance analysis of the proposed algorithm with PF, APF, PSAPF and HPSO is performed. It is demonstrated that the proposed algorithms are scalable and can recover from catastrophic failures. Whole body human tracking results have suggested that the mean error performance of the proposed algorithms is significantly better than PF, PSAPF and HPSO with different level of significance. Further, time complexity analysis

Table 8.13: Execution time of different algorithm for Lee Walk sequence S_5

Algorithm	Function Eval.	Time/Frame	Time/Frame/Function Eval.
PF	7200	204 s	28.33 ms
APF	7200	204 s	28.33 ms
PSAPF [61]	7200	120 s	16.67 ms
HPSO [61]	7200	76.8 s	10.67 ms
AP-PSO	2500	64.1 s	25.64 ms
MAB	2500	64.9 s	25.96 ms
DM-AIS	2500	66.4 s	26.56 ms
PSO-AIS	2500	65.6 s	26.24 ms

study has suggested the effectiveness of the proposed algorithms, as they use less function evaluations to achieve same or better tracking performance.

Chapter 9

Conclusions and Future Research Directions

This dissertation proposes four metaheuristic optimization algorithms to address the challenging AHT problem. Section 9.1 summarizes the characteristics of the proposed algorithms. In particular, their exploration and exploitation characteristics are compared to highlight the reasons for their success and their limitations. Conclusions about the accuracy, precision, recovery and time-complexity of the proposed algorithms is also given in this section. Section 9.2 describes the research contributions of this dissertation. Section 9.3 unfolds a number of future research venues that can be explored to enhance the accuracy, precision, robustness and real time viability of the proposed algorithms for high-dimensional video tracking applications.

9.1 Conclusions

The demand for robust, fast and accurate optimization algorithms to solve complex real world problems is growing among researchers. Many were inspired by nature to look for solutions of such complex problems. A keen observation of nature leads to the development of many important paradigms of computational intelligence in the form of swarm intelligence and evolutionary computational algorithms. These optimization algorithms operate iteratively. The individuals are driven towards the optimization goal in a guided random search using distributed structure and sharing information.

The milieu of this research is swarm intelligence and evolutionary computation. Four algorithms were proposed, each has its unique characteristics with respect to global and local solution search. These characteristics are summarized in Table 9.1.

Table 9.1: Local and global search mechanisms of the proposed algorithms

Algorithm	Local search	Global search
AP-PSO	Fitness value based adaptive parameter velocity and position update procedures	
MAB	Limited local search	Velocity and position update procedures
DM-AIS	Fitness value based cloning and secondary mutation operations	Fitness value based primary mutation operation
PSO-AIS	Fitness value based cloning and secondary mutation operations of DM-AIS	Fitness value based adaptive parameter velocity and position update procedures of AP-PSO and fitness value based primary mutation operation of DM-AIS

The first proposed algorithm, AP-PSO, is based on the social interaction of a swarm. The original PSO algorithm uses fixed inertia weight as well as social and cognitive learning factors. PSO has shown good initial convergence but struggles to converge to a local/global solution due to random velocity update mechanism. A fitness value based scheme to update the social and cognitive learning factors is proposed. This fitness adaptive parameter update mechanism has imparted exploitation characteristics to good individuals and exploration characteristics to poor ones. Inertia weight is also gradually reduced with iterations.

The second proposed algorithm, MAB algorithm, uses only social interaction for velocity update and is augmented by a local search for selected individuals. The loudness of the bat, A_i, is varied according to the fitness of the bat in the proposed algorithm contrary to linearly decreasing it in the original Bat algorithm, which results in enhancing its probability for selection for further modifications.

The third proposed algorithm, DM-AIS is based on a two-stage mutation operation. A primary mutation stage favors exploration, while a secondary mutation stage favors exploitation. A DE like differential mutation scheme is proposed in this dissertation. Individuals are selected for mutation in different manner for each of the two mutation stages. Furthermore, the scalar mutation factor is based on fitness value. A good fitness value results in a local search while a poor fitness value results in a global search.

The last proposed algorithm, PSO-AIS, is a hybridization of AP-PSO and DM-AIS algorithms that exploits the advantages of both algorithms. AP-PSO is utilized for fast initial convergence and DM-AIS is subsequently used for thorough local search in this hybridized scheme.

All the proposed algorithms adapt their solution search capability on the basis of the quality of obtained results. As a consequence, the proposed algorithms do not 'blindly' explore the entire solution space rather they utilize the individuals' 'fitness values' to locate the optimal solution. This reduces the demand for particles which grows exponentially with dimensionality for PF [87]. Hence, the 'curse of dimensionality' that plague generic PF is tackled. Moreover, since the whole swarm moves towards the optimal solution, the proposed algorithms also tackle the problem of 'degeneracy' encountered in generic PF.

Experiments on a variety of problems including parameter identification, low-dimensional state estimation, low-dimensional video tracking and high-dimensional human tracking demonstrate significant performance improvements realized by the proposed algorithms over the generic PF, PSAPF and HPSO. Further, all four proposed algorithms have shown stable performance during the course of experiments. DM-AIS has performed the best followed by PSO-AIS algorithm. AP-PSO is third in terms of performance while MAB algorithm is the last in this performance queue. A comparison of the frame-to-frame MAP tracking error of the four algorithms for Lee Walk sequence S_5 is illustrated in Figure 9.1.

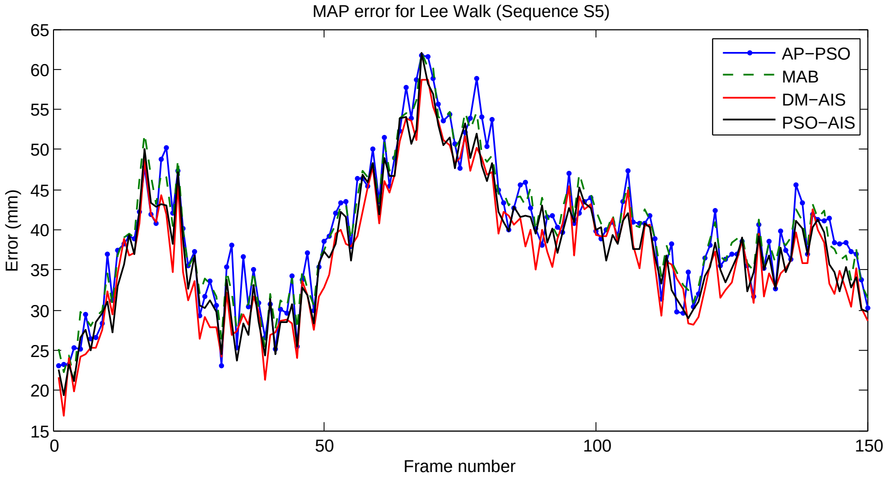


Figure 9.1: Comparison of MAP error of the proposed algorithms for Lee Walk Sequence

The success of DM-AIS algorithm in terms of least mean MAP error and standard deviations is due to the fact that the parent vector is mutated with scaled population-derived difference vector, where the computation of difference vector depends upon the mutation stage. The difference tends to adapt to the natural scaling of the problem as the generation number grows. For example, if the population converge for one of the problem variables and remain widely diverged for another variable then the difference vector sampled from this population will be small for the former variable, while it will be large for latter variable. The convergence of the algorithm is greatly enhanced due to this automatic adaptation.

A similar scale difference vector stochastic attraction is shown by AP-PSO algorithm towards global and individual's best solution and MAB algorithm towards global best solution. But both these algorithms also include a random multiplicative parameter which is not the part of DM-AIS mutation process. As a result, the proposed AP-PSO and MAB algorithms induce a random velocity effect which is good for exploration but degrades the convergence behavior in later stages. Furthermore, the absence of a strong local search capability contributes to lower performance scores.

The performance of DM-AIS and PSO-AIS algorithms are almost identical although the latter was expected to perform better due to the combined fast initial convergence of AP-PSO and the strong local search of DM-AIS. This could be a result of the use of fewer DM-AIS iterations in this hybrid scheme compared to the DM-AIS algorithm to maintain constant function evaluations for comparison. Further exploration to get insight of this behavior can also be an interesting research direction and may help understand the algorithm dynamics in a better fashion.

The scalability analysis studies, using a benchmark test function, indicated that with the increase in problem dimension, the optimization performance deteriorates. It is further noted that with the increase in function evaluation, the optimization performance improves. This improvement follows an exponential trend and is much faster than PF. The proposed algorithms also demonstrated excellent recovery characteristics in the wake of error accumulation and sudden change in motion pattern, which leads to believe that the proposed algorithms are immune to catastrophic or algorithmic failures.

Further, statistical performance analysis exhibited improved performance of the proposed algorithms over the other considered algorithms in this dissertation. Run-time computational results on whole body human tracking illustrated that although PSAPF and HPSO execute faster per function evaluation, 16.67 ms and 10.67 ms respectively compared to 26 ms for the proposed algorithms, but significantly more function evaluations per frame are required to achieve same level of accuracy as that of the proposed algorithms.

As far as selecting a particular algorithm for a certain application is concerned, it depends upon the type of application and desired level of accuracy. Applications where speed is the major concern, such as video gaming or online activity recognition, AP-PSO could be a better choice as it is easy to code and provide accurate results quickly. On the other hand, if the application requires more precise results, such as patient rehabilitation or off-line sport activity analysis, DM-AIS should be considered due to its better mean MAP tracking error and standard deviation performance among the proposed algorithms.

The presented results in various chapters of this dissertation suggest that the proposed algorithms can be applied to a wide variety of low-dimensional and high-dimensional optimization problems such as system identification, filtering, video tracking etc. The proposed algorithms can easily be scaled to any problem dimension due to their adaptive nature of parameter update procedure. Therefore, these algorithms can be applied to other engineering, science, business and economic fields.

9.2 Research Contributions

The following is a list of contributions resulting from this work:

- Proposed four optimization algorithms:
  - Swarm intelligence based AP-PSO algorithm.
  - Bat echolocation characteristic base MAB algorithm.
  - Clonal selection based DM-AIS algorithm.
  - AP-PSO and DM-AIS based hybrid PSO-AIS algorithm.
- Suggested a fitness value based update of:
  - Social and cognitive learning parameters in AP-PSO.
  - Loudness in proposed MAB algorithm.
  - Multiplicative parameter for the mutant vector in DM-AIS algorithm.
- Proposed optimization algorithms that have an inherent automatic ability to switch between exploitation and exploration.
- Reduced particle requirement for tracking/optimization from that required by PF.
- Improved computational efficiency due to reduced particle requirements.
- Alleviated 'degeneracy' due to the swarm behavior of the proposed algorithms.

- Imparted the ability to recover from catastrophic failures due to the decentralized search structure of the proposed algorithms.
- Relaxed the requirement for accurate dynamical/temporal model by embedding diversified search ability in the proposed algorithms.
- Proposed algorithms with improved performance in identification, low-dimensional tracking and high-dimensional AHT problems.
- Suggested an intuitive easy to render schematic for the probabilistic generative whole body tracking framework.

9.3 Future Research Directions

On the basis of this work, we have identified the following aspects that represent a good venue for future investigation for researchers exploring AHT and complex optimization domains.

- Visual tracking systems require an explicit or implicit model for shape and appearance of the object of interest. Absence of a suitable model limits the performance of the tracking system. Edges and silhouettes are used in this study. Use of other appearance models such as color histograms, texture and shape models such as contours or the orientation of contour edges may be explored to improve the discriminative ability of the tracking system.
- Dimension reduction techniques can discover the intrinsic structure of high-dimensional data. They are used as a pre-processing step prior to the application of complex modeling techniques. PCA and probabilistic PCA are used for human tracking [129], [130], [155]. PCA is unsuitable for non-Gaussian distributions which is normally the case in image based video tracking. The use of some dimension reduction techniques such as projection pursuit, random projections, density networks, to reduce the dimensionality of the articulated model before the application of metaheuristic optimization techniques will be an interesting venue for research to further reduce particle requirements and improve its real time applicability. However, reduction in dimensionality may effect the accuracy of tracking results which is more important in certain applications such as patient rehabilitation, sport activity analysis etc. A through analysis will be required to comprehend the effect of dimensionality reduction on tracking performance.

- The construction of a joint latent space for shape, appearance and dynamics will also be an interesting area to explore. This will simplify the mapping procedure from input to the latent space and vice versa.

- The number of particles required for tracking in a PF framework grows exponentially with dimension [87]. All PF algorithms developed to address the 'curse of dimensionality' use a fixed number of particles for likelihood approximation. When applied to the high-dimensional AHT problem, the computational cost remains very high throughout the course of tracking. The use of the probability density measure methods, such as Kullback-Leibler Distance (KLD) or Earth mover's distance, can be explored to adaptively improve the sample size which will alleviate the computational requirement as the search is approaching the optimal solution.

- A good prior dynamical motion model will provide a powerful cue in the presence of occlusion and measurement noise and can improve the convergence of the optimization method. The use of this motion model will constraint the propagation of samples to those portions of the solution space which are more likely to conform to the motion. It will reduce the number of samples needed to reach the peak of the posterior distributions. In this research work a zero velocity motion model is used for prior dynamics. The use of a more sophisticated dynamical model is expected to improve the performance of the tracking system.

- Human activity recognition is an important area of computer vision research [89, 126]. The goal of activity recognition is an automated analysis of an event from video data. Its applications include surveillance systems, patient monitoring systems, security systems etc. Recognition of most of these high-level activities consists of recognition of a series of simple actions. The predicted states obtained as the output of the metaheuristic optimization based tracking systems, developed in this research, can later be used as an input for such an activity recognition system.

- Euclidean distance is used in this research for likelihood computations. In case of decision variables with different search ranges, scale invariant Mahalanobis distance measure, which takes into account the correlations of the data set, should be used to avoid any dependence of likelihood function on the scale of variables. In case of diagonal covariance matrix a normalized Euclidean distance can be used as:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{s_i^2}} \tag{9.1}$$

where s_i is the standard deviation.

- In this research simplified metaheuristics are applied to deal with a complex AHT problem. On a pure algorithmic note, it will be interesting to develop more complex metaheuristic algorithms that can mimic the working of a complex natural system, leading to a self-regulating, self evolving, more intelligent metaheuristic that can evolve into a powerful next generation to solve complex optimization problems.

Appendix A

Edge Gradient Computation

A number of points (x_i, y_i) are selected on the predicted object. The gradient at pixel (x_i, y_i) is obtained as a maximum gradient by a local search along the normal direction as [15]:

$$g(x_i, y_i) = \max_{(x_n, y_n) \in l_n} \{g(x_n, y_n)\} \tag{A.1}$$

The gradient in x and y direction for pixel (x_n, y_n) is computed as:

$$\begin{aligned} g_x(x_n, y_n) &= I(x_{n-2}, y_n) + 2I(x_{n-1}, y_n) - 2I(x_{n+1}, y_n) - I(x_{n+2}, y_n) \\ g_y(x_n, y_n) &= I(x_n, y_{n-2}) + 2I(x_n, y_{n-1}) - 2I(x_n, y_{n+1}) - I(x_n, y_{n+2}) \end{aligned} \tag{A.2}$$

and the final gradient at point (x_n, y_n) is computed as:

$$g(x_n, y_n) = \sqrt{g_x^2(x_n, y_n) + g_y^2(x_n, y_n)} \tag{A.3}$$

The normalized sum of the gradient magnitude for all the selected points is computed to get the normalized gradient as:

$$g(\Gamma) = \frac{1}{N_\Gamma} \sum_{i=1}^{N_\Gamma} g(x_i, y_i) \tag{A.4}$$

Appendix B

Non-linear Bayesian Tracking

The Bayesian approach for state estimation provides the foundation for the derivation of PF. The probabilistic Bayesian approach calculates the posterior distribution of states conditioned on all the measurements $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$. The approach is based on Bayes' rule which is a powerful tool for inference under uncertain conditions [135].

Consider the following non-linear dynamical system equation and measurement equation:

$$\boldsymbol{x}_t = \boldsymbol{f}_t(\boldsymbol{x}_{t-1}, \boldsymbol{\mu}_t) \tag{B.1}$$

$$\boldsymbol{z}_t = \boldsymbol{h}_t(\boldsymbol{x}_t, \boldsymbol{\nu}_t) \tag{B.2}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{z}_t$ are the system states and measurements respectively and t and $t-1$ are the current and previous time steps, $\boldsymbol{f}_t$ and $\boldsymbol{h}_t$ are possible non-linear functions of the states, $\boldsymbol{\mu}_t$ and $\boldsymbol{\nu}_t$ are independent and identically distributed (i.i.d.) processes and measurement noises respectively.

A Bayesian tracking framework recursively calculates some degree of belief in the state $\boldsymbol{x}_t$ based on the given data $\boldsymbol{z}_{1:t}$; all the observations up to the current time step t [86]. In other words, it constructs the probability density function (pdf) $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$; conditional probability density of states $\boldsymbol{x}_t$ given all the observations up to the current time step t. If the posterior pdf $p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})$ at time $t-1$ is available, then the prior pdf of the states at time t can be predicted using the system model via the Chapman-Kolmogorov equation as:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{1:t-1}) \times p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1} \tag{B.3}$$

Considering system model to be a first order Markov process, above equation reduces as:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \times p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1} \tag{B.4}$$

First density inside the integral is defined by dynamical system equation (Equation B.1) and the second term inside the integral is the posterior density at time step $t-1$.

Bayes' rule can be used to find the conditional probabilities. According to it the conditional probability is given as:

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B)}$$

where $p(A)$ is the marginal probability of A and as it does not take any information about B, it is called *prior* and $p(A|B)$ is the conditional probability of A given B and as it depends/derives from the specific value of B, hence it is called *posterior*.

After obtaining the measurements $\boldsymbol{z}_t$ at time step t, the posterior density (pdf) of the state $\boldsymbol{x}_t$ is calculated by Bayes' rule as:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_{1:t}|\boldsymbol{x}_t) \times p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_{1:t})} \tag{B.5}$$

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t, \boldsymbol{z}_{1:t-1}|\boldsymbol{x}_t) \times p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_t, \boldsymbol{z}_{1:t-1})} \tag{B.6}$$

Using $p(A, B|C) = p(A|B, C) \times p(B|C)$ [102], to first term in the numerator:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{x}_t) \times p(\boldsymbol{z}_{1:t-1}|\boldsymbol{x}_t) \times p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}) \times p(\boldsymbol{z}_{1:t-1})} \tag{B.7}$$

Applying Bayes' rule to the second term in the numerator:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{x}_t) \times p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) \times p(\boldsymbol{z}_{1:t-1}) \times p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}) \times p(\boldsymbol{z}_{1:t-1}) \times p(\boldsymbol{x}_t)} \tag{B.8}$$

Utilizing the fact that observations are independent of each other and only depend on current states, the first term in the numerator simplifies. Furthermore, canceling identical terms in the numerator and denominator results in the following expression:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t) \times p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})} \tag{B.9}$$

In the above expression, the term on the left hand side of the equation is posterior density at time t, on the right hand side; the first term in the numerator is observation density, second term is the prior density at time t and the term in the denominator is the evidence/observation. The normalizing constant, denominator in the above expression, is obtained by Chapman-Kolmogorov equation as:

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}) = \int p(\boldsymbol{z}_t|\boldsymbol{x}_t) \times p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_t \tag{B.10}$$

Using Equation B.4 in Equation B.9 results in:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t)}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})} \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \times p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1} \tag{B.11}$$

In the update state, Equation B.9, the measurement $\boldsymbol{z}_t$ is used to modify the prior density to obtain the required posterior density of the current state $\boldsymbol{x}_t$. The recursive relations Equation B.4 and Equation B.9 form the basis of the optimal Bayesian solution.

B.1 Challenges in Applying Non-linear Bayesian Tracking

The Bayesian solution, Equation B.11, recursively calculates the exact posterior density [86]. This recursive propagation of the posterior density is only a conceptual solution. It can not be analytically determined for a general case. The solution is available for a restrictive set of cases, like KF, which assumes that the posterior density at every time step is Gaussian. If this assumption does not hold, then the analytical computation of the integral in Equation B.11 may become intractable. Approximate solutions such as PF are applied in this situation. The PF is a MC based simulation technique for implementing a recursive Bayesian filter. MC methods are a class of computational algorithms that rely on random samples to compute the results. The technique provides an approximate solution of a complex physical or mathematical problem. They are used in mathematics for the evaluation of complex integrals. The idea is to use random samples with associated weights to represent the posterior density. MC methods draw samples from a proposal density called 'importance density'. The optimal choice of 'importance density', which minimizes the variance of the weights, is not obvious.

The Sampling Importance Resampling (SIR) method chooses the prior density as the 'importance density' to solve this problem. This choice makes the importance density independent of the measurement $\boldsymbol{z_t}$, so state-space is explored without any knowledge of the observations. As a result, the filter is inefficient and sensitive to outliers. These Bayesian filter issues and their solutions are given in the block diagram in Figure B.1. The PF, Sequential Importance Sampling (SIS) algorithm, and the SIR method are described in details in the following sections.

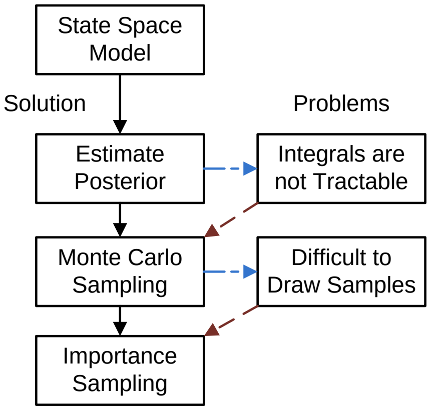


Figure B.1: Problems and solutions of Bayesian formulation

B.2 Sequential Importance Sampling (SIS) Algorithm

The SIS is a technique for implementing a recursive Bayesian filter by MC simulations [38]. The basic idea is to represent the posterior density function by a set of random samples with associated weights. The estimates are computed based on these samples and weights.

Let $S_t = \{\boldsymbol{x}^i_{0:t}, \pi^i_t\}$ be a set of particles where $\boldsymbol{x}^i_t$ is the state and π^i_t is the weight associated to the i-th particle at time instant t, and $\boldsymbol{x}_{0:t}$ is the set of all states up to time t. The posterior density at time t is approximated by:

$$p(x_{0:t}|z_{1:t}) \approx \sum_{i=1}^{N} \pi^i_t \times \delta(\boldsymbol{x}_{0:t} - \boldsymbol{x}^i_{0:t}) \tag{B.12}$$

where $\delta(\cdot)$ is the Dirac delta measure.

The expression B.12 is a discrete weighted approximation of the true posterior, $p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t})$. The weights are chosen on the principle of importance sampling [37]. It supposes that $p(x)$, which is proportional to $\tau(x)$, $p(x) \propto \tau(x)$, is a probability density from which it is difficult to draw samples but $\tau(x)$ can be evaluated. As a result $p(x)$ can also be evaluated up to proportionality. It also assumes that the samples x^i can be generated from a proposal density $q(\cdot)$ called importance density . Then the weighted approximation of $p(\cdot)$ is:

$$p(x) \approx \sum_{i=1}^{N} \pi^i \times \delta(x - x^i) \tag{B.13}$$

where π^i, which is proportional to the ratio of $\frac{\tau(x^i)}{q(x^i)}$, is the normalized weight of the i-th particle.

Therefore, if the samples $\boldsymbol{x}_{0:t}^i$ are drawn from importance density $q(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t})$ then the weights in Equation B.12 are defined as:

$$\pi_t^i \propto \frac{p(\boldsymbol{x}_{0:t}^i|\boldsymbol{z}_{1:t})}{q(\boldsymbol{x}_{0:t}^i|\boldsymbol{z}_{1:t})} \tag{B.14}$$

In sequential case, samples are available to approximate $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1})$ and from there we want to approximate $p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t})$ with a new set of samples. If we factorize importance density as:

$$q(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t}) \quad q(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1}) \tag{B.15}$$

then the samples $\boldsymbol{x}_t^i \sim q(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t})$ can be obtained by augmenting existing samples $\boldsymbol{x}_{0:t-1}^i \sim q(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1})$ with a new state $\boldsymbol{x}_t^i \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t})$. To derive the weight update equation, it is required to express $p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t})$ in terms of $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1})$, $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.

$$\begin{aligned}
p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t}) &= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t-1})p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})} \\
&= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})} \times p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1}) \qquad \text{(B.16)} \\
&= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})} \times p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1}) \\
&\propto p(\boldsymbol{z}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \times p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1}) \qquad \text{(B.17)}
\end{aligned}$$

By substituting Equation B.15 and Equation 3.17 into Equation B.14, the weight update equation can be shown to be:

$$\begin{aligned}\pi_t^i &\propto \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)p(\boldsymbol{x}_{0:t-1}^i|\boldsymbol{z}_{1:t-1})}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{0:t-1}^i,\boldsymbol{z}_{1:t})q(\boldsymbol{x}_{0:t-1}^i|\boldsymbol{z}_{1:t-1})} \\ &= \pi_{t-1}^i\frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{0:t-1}^i,\boldsymbol{z}_{1:t})} \end{aligned} \tag{B.18}$$

Furthermore, if $q(\boldsymbol{x}_t^i|\boldsymbol{x}_{0:t-1}^i,\boldsymbol{z}_{1:t}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{z}_t)$, then the importance density becomes only dependent on $\boldsymbol{x}_{t-1}$ and $\boldsymbol{z}_t$. This is useful when only a filtered estimate of $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ is required. The modified weights are given as [86]:

$$\pi_t^i \propto \pi_{t-1}^i \times \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i,\boldsymbol{z}_t)} \tag{B.19}$$

and the posterior filter density $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ is approximated as:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \approx \sum_{i=1}^{N} \pi_t^i \times \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i) \tag{B.20}$$

It can be shown that as $N \to \infty$, the approximation in Equation B.20 approaches the true posterior density $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$.

B.3 Sampling Importance Resampling (SIR) Filter

The SIR algorithm [49] can be derived from SIS algorithm by an appropriate choice of importance density and resampling step. The importance density, $q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i,\boldsymbol{z}_t)$, is chosen to be the prior density $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$. By choosing the prior density as the importance density, samples $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$ can be generated from the system equation, Equation B.1, and the weights are given by:

$$\pi_t^i \propto \pi_{t-1}^i \times p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{B.21}$$

The weights are normalized to ensure that sum of weights is equal to 1 as follows:.

$$\pi^i = \frac{\pi^i}{\sum_{j=1}^{N}\pi^j} \tag{B.22}$$

where $i = 1...N$, and N is the number of particles.

In the resampling stage, particles are selected on the basis of likelihoods, normalized weights, and new particles are generated to replace the particles with negligible weights. The systematic resampling [68] technique is used as a resampling method as:

1. Generate a random number r that is uniformly distributed on [0-1].
2. Accumulate the likelihoods into a sum, one at a time, until the accumulated sum is greater than r, i.e.,

   $$\sum_{m=1}^{j} \pi^m \geq r \tag{B.23}$$

   The new particle $\boldsymbol{x}_{t,j}^{+}$ is then defined as $\boldsymbol{x}_{t,j}^{+} = \boldsymbol{x}_{t,j}^{-}$.

Bibliography

[1] Agarwal A. and Triggs B. Tracking articulated motion with piecewise learned dynamic models. *in: European Conference on Computer Vision, Prague, Czech Republic*, May 11-14, 2004.

[2] Alwasel A., Elrayes K., Abdel-Rahman E., and Haas C. Reducing shoulder injuries among construction workers. *Gerontechnology*, 11(2):242–246, 2012.

[3] Chatterjee A. and Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & Operations Research*, 33(3):859 – 871, 2006.

[4] Eihab Abdel-Rahman and Mohamed Samir Hefzy. A two-dimensional dynamic anatomical model of the human knee joint. *Journal of biomechanical engineering*, 115(4A):357–365, 1993.

[5] Eihab Muhammed Abdel-Rahman and Mohamed Samir Hefzy. Three-dimensional dynamic behaviour of the human knee joint under impact loading. *Medical engineering & physics*, 20(4):276–290, 1998.

[6] Ahmad Abdul-Rahim, Basir Otman A, Imam Muhammad Hasan, and Hassanein Khaled. An efficient, effective, and robust decoding heuristic for metaheuristics-based layout optimization. *International journal of production research*, 44(8):1545–1567, 2006.

[7] Abdul-Rahim Ahmad, Otman A Basir, Khaled Hassanein, and Muhammad Hasan Imam. Improved placement algorithm for layout optimization. *Proc. of the 2nd Intl Industrial Engineering Conf.(IIEC04)*, 2004.

[8] Chang Wook Ahn. *Advances in Evolutionary Algorithms: Theory, Design and Practice (Studies in Computational Intelligence).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[9] Alireza Alfi. PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems. *Acta Automatica Sinica*, 37(5):541–549, 2011.

[10] Ahmad A.R. *An Intelligent Expert System for Decision Analysis & Support in Multi-Attribute Layout Optimization*. PhD thesis, University of Waterloo, 2005.

[11] Ahmad A.R., Basir O., Hassanein K., and Imam M.H. An effective module placement strategy for genetic algorithms based layout design. In *Intl Journal of Production Research, vol. 44, No. 8*, pages 1545–1567, 2006.

[12] A. O. Balan, L. Sigal, and M. J. Black. A quantitative evaluation of video-based 3d person tracking. *Proceedings of the 14th International Conference on Computer Communications and Networks*, October 15-16, 2005.

[13] Jan Bandouch, Florian Engstler, and Michael Beetz. Evaluation of hierarchical sampling strategies in 3d human pose estimation. In Mark Everingham, Chris J. Needham, and Roberto Fraile, editors, *BMVC*. British Machine Vision Association, 2008.

[14] Martin Bergtholdt, Jörg Kappes, Stefan Schmidt, and Christoph Schnörr. A study of parts-based object class detection using complete graphs. *Int. J. Comput. Vision*, 87(1-2):93–117, March 2010.

[15] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California*, pages 232–237, 1998.

[16] Christian Blum, Jakob Puchinger, Günther R. Raidl, and Andrea Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Appl. Soft Comput.*, 11(6):4135–4151, September 2011.

[17] C. Bregler and J. Malik. Tracking people with twists and exponential maps. *Proc. Conf. Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[18] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision, 56(3)*, pages 179–194, 2004.

[19] Dave Bullock and John Zelek. Towards real-time 3-D monocular visual tracking of human limbs in unconstrained environments. *Real-Time Imaging*, 11(4):323–353, August 2005.

[20] F. M. Burnet. *The clonal selection theory of acquired immunity.* Vanderbilt University Press, Cambridge, UK, 1959.

[21] Clapp T. C. *Statistical Methods for the Processing of Communication Data.* PhD thesis, University of Cambridge, Cambridge, UK, 2000.

[22] Musso C., Oudjane N., and LeGland F. Improving regularized particle filters. In *Sequential Monte Carlo methods in practice. Eds. Doucet et. al.*, New York, 2001. Springer-Verlag.

[23] JAI Pulnix Sunnyvale CA. Pulnix cameras. http://www.vicon.com/. Accessed January 10, 2014.

[24] K. Choo and D.J. Fleet. People tracking with hybrid Monte Carlo. *Proc. Intl Conf. Computer Vision*, pages 321–328, 2001.

[25] Cristbal Curio and Martin A. Giese. Combining view-based and model-based tracking of articulated human movements. In *WACV/MOTION*, pages 261–268. IEEE Computer Society, 2005.

[26] Vincenzo Cutello, Giuseppe Narzisi, Giuseppe Nicosia, and Mario Pavone. An immunological algorithm for global numerical optimization. In El-Ghazali Talbi, Pierre Liardet, Pierre Collet, Evelyne Lutton, and Marc Schoenauer, editors, *Artificial Evolution*, volume 3871 of *Lecture Notes in Computer Science*, pages 284–295. Springer, 2005.

[27] Vincenzo Cutello, Giuseppe Nicosia, and Mario Pavone. Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator. In Hisham Haddad, editor, *SAC*, pages 950–954. ACM, 2006.

[28] Jacek Dabrowski and Marek Kubale. Computer experiments with a parallel clonal selection algorithm for the graph coloring problem. In *IPDPS*, pages 1–6. IEEE, 2008.

[29] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evolutionary Computation*, 15(1):4–31, 2011.

[30] B. Daubney and Xianghua Xie. Tracking 3d human pose with large root node uncertainty. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1321–1328, June 2011.

[31] Ben Daubney, David Gibson, and Neill Campbell. Estimating pose of articulated objects using low-level motion. *Computer Vision and Image Understanding*, 116(3):330 – 346, 2012. Special issue on Semantic Understanding of Human Behaviors in Image Sequences.

[32] L. N. de Castro and F. J. Von Zuben. The clonal selection algorithm with engineering applications. *In GECCO'00, Workshop on Artificial Immune Systems and Their Applications*, pages 36–37, 2000.

[33] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 716–721 vol.2, 1999.

[34] Q. Delamarre and O. Faugeras. 3D articulated model and multiview tracking with physical forces. *Computer Vision and Image Understanding, Vol. 81*, pages 328–357, 2001.

[35] J. Deutscher, A. Davison, and I. Reid. Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. *in: Computer Vision and Pattern Recognition, Kauai Marriott, Hawaii*, December 9-14, 2001.

[36] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *in: IJCV, (61)2*, pages 185–205, 2004.

[37] A. Doucet. On sequential monte carlo methods for bayesian filtering. *Dept. Eng., Univ. Cambridge, UK, Tech. Rep.*, 1998.

[38] Arnaud Doucet, Nando De Freitas, and Neil Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York, 2001.

[39] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61(1):55–79, January 2005.

[40] Juergen Gall, Bodo Rosenhahn, and Hans-Peter Seidel. Clustered stochastic optimization for object recognition and pose estimation. In Fred A. Hamprecht, Christoph Schnrr, and Bernd Jhne, editors, *DAGM-Symposium*, volume 4713 of *Lecture Notes in Computer Science*, pages 32–41. Springer, 2007.

[41] Jürgen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. Optimization and filtering for human motion capture - a multi-layer framework. *International Journal of Computer Vision*, 87(1-2):75–92, 2010.

[42] Amir H. Gandomi and Xin-She Yang. Chaotic bat algorithm. *Journal of Computational Science*, (0):–, 2013.

[43] Simon M. Garrett. Parameter-free, adaptive clonal selection. In *Congress on Evolutionary Computing (CEC 2004)*, volume 1. IEEE Press, USA, 2004.

[44] D.M. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding 73 (1)*, pages 82–98, 1999.

[45] Hong-Wei Ge and Yan-Chun Liang. A hidden markov model and immune particle swarm optimization-based algorithm for multiple sequence alignment. In *Australian Conference on Artificial Intelligence*, pages 756–765, 2005.

[46] Hong-Wei Ge, Liang Sun, Yanchun Liang, and Feng Qian. An effective pso and ais-based hybrid intelligent algorithm for job-shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 38(2):358–368, 2008.

[47] Maoguo Gong, Licheng Jiao, Lining Zhang, and Wenping Ma. Improved real-valued clonal selection algorithm based on a novel mutation method. In *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, pages 662–665, 2007.

[48] Maoguo Gong, Lining Zhang, Licheng Jiao, and Wenping Ma. Differential immune clonal selection algorithm. In *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, pages 666–669, 2007.

[49] N. Gordon, D. Salmond, and A.F.M Smith. Novel approach to non-linear and non-gaussian bayesian state estimation. *Proc. Inst. Elect. Eng., 140*, pages 107–113, 1993.

[50] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, August 2004.

[51] Ali B Hashemi and MR Meybodi. Adaptive parameter selection scheme for PSO: A learning automata approach. In *Computer Conference, 2009. CSICC 2009. 14th International CSI*, pages 403–411. IEEE, 2009.

[52] J. Heikkil and O. Silvn. A four-step camera calibration procedure with implicit image correction. *in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), San Juan, Puerto Rico*, pages 1106–1112, 1997.

[53] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing 1 (1)*, pages 5–20, 1983.

[54] IO Industries. Advanced imaging. http://www.ioindustries.com/. Accessed January 10, 2014.

[55] Adiel Ismail and Andries Petrus Engelbrecht. Self-adaptive particle swarm optimization. In Lam Thu Bui, Yew-Soon Ong, Nguyen Xuan Hoai, Hisao Ishibuchi, and Ponnuthurai Nagaratnam Suganthan, editors, *SEAL*, volume 7673 of *Lecture Notes in Computer Science*, pages 228–237. Springer, 2012.

[56] Nobuhiro Iwasaki, Keiichiro Yasuda, and Genki Ueno. Dynamic parameter tuning of particle swarm optimization. *IEEJ Transactions on Electrical and Electronic Engineering*, 1(4):353–363, 2006.

[57] Huan Chen Jian Xie, Yongquan Zhou. A novel bat algorithm based on differential operator and lévy flights trajectory, 2013.

[58] Bin Jiao, Zhigang Lian, and Xingsheng Gu. A dynamic inertia weight particle swarm optimization algorithm. *Chaos, Solitons & Fractals*, 37(3):698 – 705, 2008.

[59] Agarwal J.K. and Cai Q. Human motion analysis: a review. *Computer Vision and Image Understanding 73 (3)*, pages 428–440, 1999.

[60] V. John, S. Ivekovic, and E. Trucco. Articulated human motion tracking with HPSO. In *VISSAPP (1)*, pages 531–538, 2009.

[61] V. John, E. Trucco, and S. Ivekovic. Markerless human articulated tracking using hierarchical particle swarm optimisation. *Image Vision Comput.*, 28(11):1530–1547, November 2010.

[62] Yau-Tarng Juang, Shen-Lung Tung, and Hung-Chih Chiu. Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. *Information Sciences*, 181(20):4539 – 4549, 2011. Special Issue on Interpretable Fuzzy Systems.

[63] I. A. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. *in Conference on Computer Vision and Pattern Recognition*, pages 199–202, 1996.

[64] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. In *Transactions of the ASME–Journal of Basic Engineering*, volume 82, pages 35–45, 1960.

[65] James Kennedy and Russell C. Eberhart. *The particle swarm: social adaptation in information-processing systems*, pages 379–388. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.

[66] K. Khan, A. Nikov, and A. Sahai. A fuzzy bat clustering method for ergonomic screening of office workplaces. *Advances in Intelligent and Soft Computing*, 101:59–66, 2011.

[67] Y. S. Kim and K. S. Hong. An IMM algorithm for tracking maneuvering vehicles in an adaptive cruise control environment. *Int. J. Control Autom. and Syst.*, 2(3):310–318, Sep. 2004.

[68] G. Kitagawa. Monte carlo filter and smoother for non-gaussian non-linear state space models. *J. Comput. Graph. Statist.*, 5(1):1–25, 1996.

[69] Tomoaki Kobayashi, Keita Nakagawa, Joe Imae, and Guisheng Zhai. Real time object tracking on video image sequence using particle swarm optimization. *International Conference on Control, Automation and Systems 2007, Oct. 17-20,2007 in COEX, Seoul, Korea.*

[70] G. Komarasamy and A. Wahi. An optimized k-means clustering techniqueusing bat algorithm. *European J. Scientific Research*, 84(2):263–273, 2012.

[71] Bogdan Kwolek, Tomasz Krzeszowski, and Konrad Wojciechowski. Swarm intelligence based searching schemes for articulated 3D body motion tracking. In *Proceedings of the 13th international conference on Advanced concepts for intelligent vision systems*, ACIVS'11, pages 115–126, Berlin, Heidelberg, 2011. Springer-Verlag.

[72] Xiangyang Lan and Daniel P. Huttenlocher. Beyond trees: Common factor models for 2d human pose recovery. In *In ICCV*, pages 470–477, 2005.

[73] Neil Lawrence. Gaussian Process Latent Variable Models for Visualization of High Dimensional Data. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2004.

[74] A. Leardini, L. Astolfi, S. Fantozzi, M. Viceconti, M. G. Benedetti, and F. Catani. Advanced multimodal visualisation of clinical gait and fluoroscopy analyses in the assessment of total knee replacement. *Comput. Methods Prog. Biomed.*, 79(3):227–240, September 2005.

[75] Mun Wai Lee and I. Cohen. Proposal maps driven MCMC for estimating human body pose in static images. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–334–II–341 Vol.2, June 2004.

[76] Hui-Rong Li and Yue-Lin Gao. Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation. In *Information and Computing Science, 2009. ICIC '09. Second International Conference on*, volume 1, pages 66–69, May 2009.

[77] R. Li, M. H. Yang, S. Sclaroff, and T. P. Tian. Monocular tracking of 3D human motion with a coordinated mixture of factor analyzers. *in Proceedings of European Conference on Computer Vision*, pages 137–150, 2006.

[78] Rui Li, Tai-Peng Tian, Stan Sclaroff, and Ming-Hsuan Yang. 3d human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, 87(1-2):170–190, 2010.

[79] X. Li, H. Xu, and Z. Cheng. One immune simplex particle swarm optimization and its application. *in: Proceedings of the Fourth International Conference on Natural Computation*, pages 331–335, 2008.

[80] Cheng-Jian Lin, Cheng-Hung Chen, and Chi-Yung Lee. Efficient immune-based particle swarm optimization learning for neuro-fuzzy networks design. In *J. Inf. Sci. Eng.*, volume 24, pages 1505–1520, 2008.

[81] J.H. Lin, C.W. Chou, C.H. Yang, and H.L. Tsai. A chaotic levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. *J. Computer and Information Technology*, 2(2):56–63, 2012.

[82] Ruochen Liu, Licheng Jiao, and Haifeng Du. Adaptive immune clonal strategy algorithm. In *Proceedings. ICSP '04. 2004 7th International Conference on Signal Processing*, volume 2. Proceedings. ICSP '04. 2004 7th International Conference on, USA, 2004.

[83] Z. Liu, K.T. Ng, S.C. Chan, and X.-W. Song. A new multi-view articulated human motion tracking algorithm with improved silhouette extraction and view adaptive fusion. pages 713–716, 2013. cited By (since 1996)0.

[84] Hong Lu and Mu Zhichun. A clonal chaos adjustment algorithm for multi-modal function optimization. In *Control Conference, 2008. CCC 2008. 27th Chinese*, pages 98–102, 2008.

[85] Yanping Lv, Shaozi Li, Shuili Chen, Qingshan Jiang, and Wenzhong Guo. Particle swarm optimization based on information diffusion and clonal selection. In *SEAL*, pages 521–528, 2006.

[86] Arulampalam M. S., Maskell S., Gordom N., and Clapp T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. In *IEEE Transactions Signal Processing, vol. 50, No. 2*, pages 8–15, Feb 2002.

[87] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In David Vernon, editor, *ECCV (2)*, volume 1843 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2000.

[88] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Articulated body posture estimation from multi-camera voxel data. *in: Computer Vision and Pattern Recognition, Kauai Marriott, Hawaii*, pages 559–566, December 9-14, 2001.

[89] Ben W Miners, Otman A Basir, and Mohamed S Kamel. Understanding hand gestures using approximate graph matching. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(2):239–248, 2005.

[90] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding 81 (3)*, pages 231–268, 2001.

[91] T.B. Moeslund, A. Hiltion, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding 104 (2)*, pages 90–126, 2006.

[92] Idel Montalvo, Joaqun Izquierdo, Rafael Prez-Garca, and Manuel Herrera. Improved performance of pso with self-adaptive parameters for computing the optimal design of water supply systems. *Eng. Appl. of AI*, 23(5):727–735, 2010.

[93] D.C. Montgomery and G.C. Runger. *Applied Statistics and Probability for Engineers.* John Wiley & Sons, 2010.

[94] Kooksang Moon and Vladimir Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. In *CVPR (1)*, pages 198–205. IEEE Computer Society, 2006.

[95] A Moutzouris, J.M. del Rincon, J.-C. Nebel, and D. Makris. Human pose tracking by hierarchical manifold searching. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 866–869, Nov 2012.

[96] Ramanan Navaratnam, Arasanathan Thayananthan, Philip H. S. Torr, and Roberto Cipolla. Hierarchical part-based human body pose estimation. In William F. Clocksin, Andrew W. Fitzgibbon, and Philip H. S. Torr, editors, *BMVC*. British Machine Vision Association, 2005.

[97] Samantha Ng, Adel H. Fakih, Adam Fourney, Pascal Poupart, and John S. Zelek. Probabilistic 3d tracking: Rollator users' leg pose from coronal images. In *CRV*, pages 53–60. IEEE Computer Society, 2009.

[98] X.S. Nguyen, S. Dubuisson, and C. Gonzales. Hierarchical annealed particle swarm optimization for articulated object tracking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8047 LNCS(PART 1):319–326, 2013. cited By (since 1996)0.

[99] Hui Niu, Yongshou Dai, and Xing Peng. Particle swarm optimization with adaptive parameters and boundary constraints. *I.J. Engineering and Manufacturing*, 4:19–28, 2012.

[100] G. J. V. Nossal. Life, death and the immune system. *Scientific American, 269(3)*, pages 21–30, 1993.

[101] P. O'Donoghue. *Research Methods for Sports Performance Analysis*. Research Methods for Sports Performance Analysis. Taylor & Francis Group, 2010.

[102] A. Papoulis, editor. *Probability Random Variables and Stochastics Processes*, volume 2nd Ed. McGraw-Hill, 1984.

[103] V. Pavlovic, J. M. Rehg, and T. J. Cham. A dynamic bayesian network approach to tracking using learned switching dynamic models. *in: Proceedings of International Workshop on Hybrid Systems: Computation and Control*, pages 366–380, 2000.

[104] F.J. Perales and E.R. Hancock, editors. *Human Body model acquisition and motion capture using voxel data*, volume 2492 of *AMDO 2002*. Springer, New York, 2002.

[105] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, June 1999.

[106] R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *In ICCV*, pages 394–401, 2001.

[107] R. Plankers and P. Fua. Articulated soft objects for multi-view shape and motion capture. *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 9*, pages 1182–1187, 2003.

[108] Xing QuanZuo and Shi YongLi. The chaos artificial immune algorithm and its application to rbf neuro-fuzzy controller design. In *SMC*, pages 2809–2814. IEEE, 2003.

[109] Giilks W. R. and Berzuini C. Following a moving targtet - Monte Carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society*, 63(B):127–146, 2001.

[110] Andrew Rae and Otman Basir. A framework for visual position estimation for motor vehicles. In *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on*, pages 223–228. IEEE, 2007.

[111] Andrew Rae, Alaa Khamis, Otman Basir, and Mohamed Kamel. Particle filtering for bearing-only audio-visual speaker detection and tracking. In *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, pages 1–6. IEEE, 2009.

[112] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. *Proc. Conf. Computer Vision and Pattern Recognition*, pages 467–474, 2003.

[113] L. Raskin, M. Rudzsky, and E. Rivlin. H-apf: Using hierarchical representation of human body for 3-d articulated tracking and action classification. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 452–459, Sept 2009.

[114] Leonid Raskin, Michael Rudzsky, and Ehud Rivlin. Dimensionality reduction using a gaussian process annealed particle filter for tracking and classification of articulated body motions. *Computer Vision and Image Understanding*, 115(4):503 – 519, 2011.

[115] A. Ratnaweera, S. Halgamuge, and H.C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *Evolutionary Computation, IEEE Transactions on*, 8(3):240–255, June 2004.

[116] P. Richardson. The secrete life of bats. http://www.nhm.ac.uk/nature-online/life/mammals/bats/. Accessed May 31, 2012.

[117] P. Richardson. *Bats.* Natural History Museum, London, UK, 2002.

[118] B. Rosenhahn, T. Brox, and H.-P. Seidel. Scaled motion dynamics for markerless motion capture, Jun. 2007.

[119] Bodo Rosenhahn, Reinhard Klette, and Dimitris Metaxas, editors. *Human Motion - Understanding, Modeling, Capture and Animation*, volume 36 of *Computational Imaging and Vision.* Springer, Dordrecht, The Netherlands, 2008.

[120] L. Rouchen, D. Haifeng, and J. Licheng. Immunity clonal strategies. *Proceedings of Fifth International Conference on Computational Intelligence and Multimedia applications (ICCIMA03)*, pages 290–295, September, 2003.

[121] Van Merwe Rudolph and Doucet Arnaud. The unscented particle filter. *Department of Engineering, Cambridge University: Technical Report*, 2000.

[122] S. Saini, D.R. Bt Awang Rambli, S. Bt Sulaiman, and M.N.B. Zakaria. A study of stochastic algorithms for 3d articulated human body tracking. In *Image Information Processing (ICIIP), 2013 IEEE Second International Conference on*, pages 84–89, Dec 2013.

[123] M. Salzmann and R. Urtasun. Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 647–654, June 2010.

[124] J. Schmidt, J. Fritsch, and B. Kwolek. Kernel particle filter for real-time 3D body tracking in monocular color images. *In Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 567–572, 2006.

[125] S. Sedai, M. Bennamoun, and D.Q. Huynh. A gaussian process guided particle filter for tracking 3d human pose in video. *Image Processing, IEEE Transactions on*, 22(11):4286–4300, Nov 2013.

[126] A.-H. Shabani, J.S. Zelek, and D.A. Clausi. Human action recognition using salient opponent-based motion features. In *Computer and Robot Vision (CRV), 2010 Canadian Conference on*, pages 362–369, May 2010.

[127] H. Sidenbladh and M.J. Black. Learning image statistics for bayesian tracking. *in: International Conference on Computer Vision, Vancouver, Canada*, pages 709–716, July 9-12, 2001.

[128] H. Sidenbladh, F. DeLaTorre, and M. Black. A framework for modeling the appearance of 3D articulated figures. *In Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR'00),Grenoble, France*, pages 559–566, March 2000.

[129] Hedvig Sidenbladh, Michael J. Black, and D. J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *In European Conference on Computer Vision*, pages 702–718, 2000.

[130] Hedvig Sidenbladh, Michael J. Black, and Leonid Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *ECCV (1)*, volume 2350 of *Lecture Notes in Computer Science*, pages 784–800. Springer, 2002.

[131] L. Sigal, S. Bhatia, S. Roth, M.J. Black, and M. Isard. Tracking loose-limbed people. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–421–I–428 Vol.1, June 2004.

[132] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *Int. J. Comput. Vision*, 87(1-2):4–27, March 2010.

[133] Leonid Sigal, Michael Isard, Horst W. Haussecker, and Michael J. Black. Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 98(1):15–48, 2012.

[134] Leonid Sigal, Michael Isard, Benjamin H. Sigelman, and Michael J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *in NIPS*, pages 1539–1546. MIT Press, 2003.

[135] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons Inc., New Jersey, USA, 2006.

[136] Cristian Sminchisescu and Allan D. Jepson. Variational mixture smoothing for nonlinear dynamical systems. In *CVPR (2)*, pages 608–615, 2004.

[137] Cristian Sminchisescu, Atul Kanaujia, and Dimitris N. Metaxas. Learning joint top-down and bottom-up processes for 3D visual inference. In *CVPR (2)*, pages 1743–1752, 2006.

[138] Cristian Sminchisescu and Bill Triggs. Covariance scaled sampling for monocular 3D body tracking. In *CVPR (1)*, pages 447–454, 2001.

[139] Cristian Sminchisescu and Bill Triggs. Kinematic jump processes for monocular 3D human tracking. In *CVPR (1)*, pages 69–76, 2003.

[140] Rainer Storn and Kenneth Price. Differential evolution &ndash; a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, December 1997.

[141] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, TR-95-012, March 1995.

[142] Cham T. and Rehg J. A multiple hypothesis approach for figure tracking. *Proc. CVPR (2)*, pages 239–245, 1999.

[143] Higuchi T. Monte Carlo filter using the genetic algorithm operators. *Journal of Statistics Computation and Simulation*, 59:1–23, 1997.

[144] El-Ghazali Talbi, editor. *Metaheuristics. From Design to Implementation.* John Wiley & Sons Inc., Hoboken, New Jersey, 2009.

[145] Yang Tang, Zidong Wang, and Jian an Fang. Feedback learning particle swarm optimization. *Applied Soft Computing*, 11(8):4713 – 4725, 2011.

[146] Graham W. Taylor and Geoffrey E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1025–1032, New York, NY, USA, 2009. ACM.

[147] Radha Thangaraj, Millie Pant, Ajith Abraham, and Pascal Bouvry. Particle swarm optimization: Hybridization perspectives and experimental illustrations. *Applied Mathematics and Computation*, 217(12):5208–5226, 2011.

[148] Arasanathan Thayananthan, Björn Stenger, Philip H. S. Torr, and Roberto Cipolla. Learning a kinematic prior for tree-based filtering. In *BMVC*, pages 1–10, 2003.

[149] Tai-Peng Tian, Rui Li, and Stan Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 0:50, 2005.

[150] Tai-Peng Tian and Stan Sclaroff. Fast globally optimal 2D human detection with loopy graph models. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:81–88, 2010.

[151] Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay, and Sankar Kumar Pal. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. *Information Sciences*, 177(22):5033 – 5049, 2007.

[152] Peter Troma and Csaba Szepesvari. SLS-N-IPS: an improvement of particle filters by means of local search. *In Proc. Non-Linear Control Systems (NOLCOS01)*, 2001.

[153] R. Y. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *in: IEEE J. Robotics Automat, RA-3(4)*, pages 323–344, 1987.

[154] Vicon Motion Systems Ltd. UK. Vicon motion capture. http://www.vicon.com/. Accessed January 10, 2014.

[155] R. Urtasun, D.J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. *in: International Conference on Computer Vision, Beijing, China*, October 15-21, 2005.

[156] Raquel Urtasun, David J. Fleet, and Pascal Fua. 3D people tracking with gaussian process dynamical models. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 238–245, Washington, DC, USA, 2006. IEEE Computer Society.

[157] Raquel Urtasun and Pascal Fua. 3D human body tracking using deterministic temporal motion models. In Toms Pajdla and Jiri Matas, editors, *ECCV (3)*, volume 3023 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2004.

[158] Zhang J. W. and G. G. Wang. Image matching using a bat algorithm with mutation. *Applied Mechanics and Materials*, 203(1):88–93, 2012.

[159] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang. A bat algorithm with mutation for UCAV path planning. *The Scientific World Journal*, 2012.

[160] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. *Pattern Recognition 36 (3)*, pages 585–601, 2003.

[161] Jin Wang. Particle swarm optimization with adaptive parameter control and opposition. In *Journal of Computational Information Systems*, volume 7: 12, pages 4463–4470, 2011.

[162] Qiaoling Wang, Changhong Wang, and X. Z. Gao. A hybrid optimization algorithm based on clonal selection principle and particle swarm intelligence. In *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications - Volume 02*, ISDA '06, pages 975–979, Washington, DC, USA, 2006. IEEE Computer Society.

[163] X. Wang, W. Wan, X. Zhang, and X. Yu. Annealed particle filter based on particle swarm optimization for articulated three-dimensional human motion tracking. *Optical Engineering*, 49(1):017204, January 2010.

[164] Jui-Yu Wu. Hybrid artificial immune algorithm and particle swarm optimization for solving unconstrained global optimization problems. *2012 International Symposium on Computer, Consumer and Control (is3c)*, pages 40–43, 2012.

[165] Wang X., Zou X., Wan W., and Yu X. Articulated 3D human pose estimation with particle filter based particle swarm optimization. In *ICALIP 2010 - 2010 International Conference on Audio, Language and Image Processing, Proceedings*, pages 1094–1099, 2010.

[166] Gang Xu. An adaptive parameter tuning of particle swarm optimization algorithm. *Applied Mathematics and Computation*, 219(9):4560 – 4569, 2013.

[167] Qiu Xueni. *Artificial Immune Systems for Job Shop Scheduling Problems.* PhD thesis, The University of Hong Kong (Pokfulam, Hong Kong), 2012.

[168] X.-S. Yang. A new metaheuristic bat-inspired algorithm. *Science*, 284:65–74, 2010.

[169] Angela Yao, Juergen Gall, Luc V. Gool, and Raquel Urtasun. Learning probabilistic non-linear latent variable models for tracking complex activities. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1359–1367. Curran Associates, Inc., 2011.

[170] D. F. W. Yap, S. P. Koh, S. K. Tiong, and S. K. Prajindra. Particle swarm based artificial immune system for multimodal function optimization and engineering application problem. *Trends in Applied Sciences Research (3)*, 6:282–293, 2011.

[171] Y. Yu and C. Hou. A clonal selection algorithm by using learning operator. In *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, pages 26–29, August 2004.

[172] Z. Zhang and H.S. Seah. Real-time tracking of unconstrained full-body motion using Niching swarm filtering combined with local optimization. In *HAU3D11*, pages 23–28, 2011.

[173] Z. Zhang, H.S. Seah, C.K. Quah, and J. Sun. Gpu-accelerated real-time tracking of full-body motion with multi-layer search. *IEEE Transactions on Multimedia*, 15(1):106–119, 2013. cited By (since 1996)1.