M. Tech. (Computer Science) Dissertation Series

Critical Point Analysis of The Modified Particle Swarm Optimization Model

A dissertation submitted in partial fulfillment of the requirements of
M.Tech (Computer Science)
degree of Indian Statistical Institute, Kolkata.

by

Arindam Pal

under the supervision of

Prof. Swagatam Das

Electronics and Communication Sciences Unit
Indian Statistical Institute, Kolkata
India
July 13, 2015

Declaration

I, **Arindam Pal** , **CS1310**,registered as a student of **M. Tech (CS)**,**Indian Statistical Institute**, **Kolkata** do hereby submit my Dissertation Report entitled "**Critical Point Analysis of The Modified Particle Swarm Optimization Model**". I certify

1. The work contained in this Dissertation Report is original and has been done by me under the guidance of my supervisor.
2. The material contained in this Dissertation Report has not been submitted to any University or Institute for the award of any degree.
3. I followed by guidelines provided by the Institute in preparing the report.
4. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of report and giving their details in the bibliography.

Arindam Pal

CS1310

July 13, 2015

Acknowledgments

First and foremost I would like to thank my mentor **Prof. Swagatam Das**, ECSU , ISI Kolkata. He is an idol of mine from my adolescent days and source of constant inspiration. I thank him for treating all my mistakes and mischief with great affection and show me the way to success. All I wish in life , is to be someone like him.

I would also like to thank Prof. Pinakpani Pal, ECSU , ISI Kolkata, who evaluated my work during the mid term evaluation. He is a man of unlimited energy and it is his remarks and constant support which helped me overcome all the stress.

I would also like to thank ISI Kolkata, for providing me the opportunity to be acquainted with so many great academicians and be inspired from them. I also thank my classmates and family for supporting me all the way I needed. This is some text to be centred vertically.

Ceritificate of Approval

This is to certify that this thesis titled **”Critical Point Analysis of Particle Swarm Optimization”** submitted by Arindam Pal , embodies the work done under my supervision.

Dr. Swagatam Das
Electronics and Communication Sciences Unit,
Indian Statistical Institute,
Kolkata - 700 108

Abstract

Although different types of evolutionary algorithms for optimization exist which work efficiently, little is known about their mathematical foundations. This current work tries to gain an insight over that which will help understanding the inherent structure, thereby devising more and better algorithms. This case can be compared to the case of Strassen's Algorithm for Matrix Multiplications, which is recently found to be only one case of a broader algorithm.

The current work shows that a modified model of PSO dynamics can be linearized and the behavior of the linearized model around the equilibrium point, is analyzed via simulations. A solution to the non linear dynamics of PSO is also obtained. And calculation of settling time for the linearized model is also obtained.

Contents

List of Figures

Chapter 1

Introduction

1.1 Prologue

Throughout our lives, we make decisions. In almost all the cases , the goal of the decisions we take, is to maximize our gains and minimize costs. These kind of scenarios give birth to a particular kind of problems, namely optimization problems. Although there are several kind of existing techniques for solving these kind of problems, e.g. gradient descent , simplex (for linear optimization problems) etc , Evolutionary algorithms are new players in the market and has shown extremely good performance for different kind of difficult problems. Particle Swarm Optimization is one such evolutionary algorithm which employ several candidate solutions and pick the best one from them, through extensive search of the feasibility region of the problem. Although there exists a plethora of research work going on all over the world showing extremely good results, the mathematical modeling of these class of algorithms is still not well established. The present work deals with the mathematical formulation of such algorithms and sees the Evolutionary algorithms from a viewpoint based on Control Systems.

1.2 Optimization Problems

Optimization problems involve selection of the best element , depending on some criteria, from some set of available alternatives.

An optimization problem may be represented as:

$$\min_x (f(x) : A \to \Re) \tag{1.1}$$

where,

$$A = \{x | g_i(x) \leq a_i\}$$

A formulation like the above is called an optimization problem OR a mathematical programming problem. Here x is a point in n-dimensional Euclidean space i.e. a vector. The function f in this case is called the objective function , or in case of minimization, loss or cost function or in case of maximization, called utility or fitness function, or in certain cases , energy function or energy functional.

The inequalities $g_i(x) \leq a_i$ are called the constraints. The possible candidates which satisfy the constraints or criteria are said to be feasible solution and the ensemble of them , is called the feasible set. Thus, a point in that set is called a feasible solution.

A feasible solution that maximizes (minimizes) the objective function, is called an optimal solution.

1.3 Convex Optimization

A convex set S is a set of real numbers which obeys the following property

$$\sum_{i=1}^{n} u_i \lambda_i \in S$$

where,

$$r > 1$$

$$u_1, u_2, \ldots, u_n \in S$$

$$\sum_{i=1}^{n} \lambda_i = 1$$

$$\lambda_i > 0$$

And a function $f : S \rightarrow \Re$ is convex if it follows the following

$$f(\lambda_1 s_1 + \lambda_2 s_2 + \cdots + \lambda_n s_n) \geq \lambda_1 f(s_1) + \lambda_2 f(s_2) + \cdots + \lambda_n f(s_n) \tag{1.2}$$

where

$$\begin{aligned} & s_1, s_2 \ldots s_n \in S \\ & \sum_{i=1}^{n} \lambda_i = 1 \\ & \lambda_i > 0 \end{aligned}$$

So, convex optimization deals with the problems where both the objective and constraints are convex functions. Many of the real life problems do not involve convex optimization problems. Thus most of the nature inspired algorithms are primarily concentrated on non convex optimization problems.

1.4 Stochastic Optimization

Stochastic Optimization problems are those which involve random variables in their procedures. The random variables can either appear in the formulation of objective functions, or even in the constraints or maybe both. Some methods involve randomness in the search process to accelerate performance. These methods include simulated annealing[21] , swarm algorithms and genetic algorithms [14].

1.5 Metaheuristics

A metaheuristic is a procedure which helps to find a partial search method for providing a sufficiently good solution of an optimization problem. They are most helpful in case of problems where the information about the problem is incomplete or the computational resources are pretty limited. A special category of metaheuristics is where the collective behavior of a population of agents are considered. This category of metaheuristics is called Swarm Intelligence.

1.6 Swarm Algorithms

Swarm algorithms employ the idea of a population of droids working while interacting with each other , and also the environment. The behavior of the droids are generally random , and they follow very simple rules. Example of Swarm Algorithms are ant colonies, flocks of birds, schools of fishes etc. Particle Swarm Optimization is an example of a swarm algorithms.

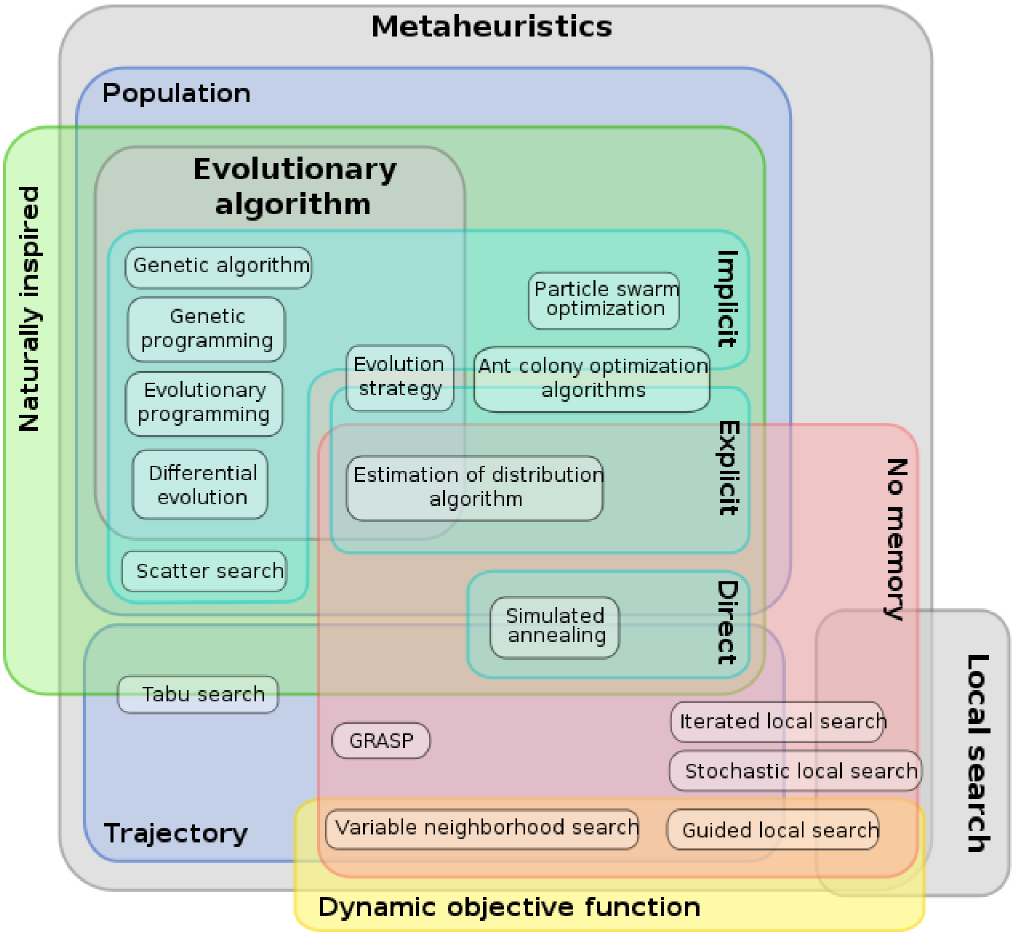


Figure 1.1: Different type of optimization algorithms[1]

1.7 State Variables and State Equations

Dynamical Systems are generally modeled by a finite number of coupled 1st order Differential Equations i.e.

$$
\begin{aligned}
\dot{x_1} &= f_1(t, x_1, x_2, \dots u_1, u_2 \dots) \\
\dot{x_2} &= f_2(t, x_1, x_2, \dots u_1, u_2 \dots) \\
&\vdots \\
\dot{x_n} &= f_n(t, x_1, x_2, \dots u_1, u_2 \dots)
\end{aligned}
$$

where,

$$
\begin{aligned}
\dot{x}_i &= \frac{dx_i}{dt} \\
u_i &= \text{input variables}
\end{aligned}
$$

We call the variables x_i State Variables and they are associated with the memory the dynamical system has of its past.

Using Vector notations to represent these equations in compact form,

$$\dot{x} = f(t, x, u) \tag{1.3}$$

where,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$$

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \cdots \\ u_n \end{bmatrix}$$

$$f(t,x,u) = \begin{bmatrix} f_1(t,x,u) \\ f_2(t,x,u) \\ \cdots \\ f_n(t,x,u) \end{bmatrix}$$

We call the equation above the State Equation and x , the state variable and u the input. The state of a system, thus means the set of real numbers required to completely specify it in the state space.

In some cases, another equation,

$$y = h(t,x,u) \tag{1.4}$$

is associated with the 1.3, which defines an output vector that consists of those particular variables according to particular interests. Which means, choice of output variables is not fixed and can be done according to the need of the problem.

We call equation 1.4 an output equation and equations 1.3 & 1.4 together as state space model, or simply the state model.

1.8 Autonomous Systems

If the state equation exists without explicit presence of an input u , which is

$$\dot{x} = f(t,x) \tag{1.5}$$

we call equation 1.5 the unforced state equation.

An unforced state equation does not mean, that input to the system occurs with a value of zero, which means no input is there. This may so happen that the input is a function of time, state variables , or even both.

Let

$$u = g(t, x)$$

Substituting u in equation 1.3 ,

$$\dot{(x)} = f(t, x, u)$$

$$\Rightarrow \dot{x} = f(t, x, g(t, x))$$

$$\Rightarrow \dot{x} = \phi(t, x) \tag{1.6}$$

A special case of equation 1.6 arises when the function ϕ doesn't depend explicitly on time. i.e.

$$\dot{x} = \phi(x) \tag{1.7}$$

In this case, the system is said to be autonomous or time invariant.

Thus, an autonomous system is a dynamical system that doesn't depend on time and also not explicitly on the input to the system. It does, however, solely depend on the system.

1.9 Equilibrium Points

One of the most important concepts in case of state equations is of equilibrium points. A point $x = x^*$ is said to be an equilibriuk point of $\dot{x} = f(t, x)$ if whenever the state of the system starts at $x = x^*$, it will remain at x^* for all future time. In case of $\dot{x} = f(x)$, the equilibrium points are real roots of the equation $f(x) = 0$.

An equilibrium point can be isolated, i.e there are no other equilibrium points in its vicinity, or there could be a continuum of equilibrium points.

1.10 Introduction to PSO

Kennedy and Eberhart [18] introduced the concept of function-optimization by means of a particle swarm in 1995. Currently the basic PSO and its variants constitute one of the most well-known families of global optimizers over real parameter space. In PSO, each trial solution is modeled as a particle and several such particles collectively form a swarm. Particles fly through the multi-dimensional search space following a typical dynamics in search of the global optima. At any particular instance, each particle has a position and a velocity. At the beginning, a population of particles is initialized with random position-vectors marked by $x_i(0)$, and random velocities as $v_i(0)$. Each particle in the swarm adapts its search pattern by learning from its own experience as well as from other particles'. A particle has the tendency to move towards a better search area with a definite velocity determined by the information collected by its own self and the other members of the swarm over the course of the search process.At time step $t + 1$, the algorithm updates the positions as

$$v_i(t+1) = \omega.v_i(t) + a.rand_1.(l_i(t) - x_i(t)) + b.rand_2.(g(t) - x_i(t))$$
$$x_i(t+1) = x_i(t) + v_i(t+1)$$

where,

$\omega =$ inertia factor

$l_i(t) =$ best position observed by ith particle

$g(t) =$ best position observed by the swarm of particles

$a, b =$ acceleration coefficients reflecting the weighting of stochastic acceleration terms that pull each particle toward particle best and global best positions

The first term of the velocity updation formula represents the inertial velocity of the particle. The attractors $l_i(t)$ and $g(t)$ are not fixed positions. Rather they are also updated through the iterations.

1.10.1 Nature of PSO parameters:

The terms a and b are termed as “self-confidence” and “swarm confidence”. These terminologies are due to Venter and Sobieski[33].These terminologies provide an insight from a sociological standpoint. Since the coefficient a has a contribution towards the self-exploration (or experience) of a particle, we regard it as the particle’s self-confidence. On the other hand, the coefficient b has a contribution towards motion of the particles in global direction, which takes into account the motion of all the particles in the preceding program iterations, naturally its definition as “swarm confidence” is apparent.

1.10.2 Random numbers:

$rand_1$ and $rand_2$ stand for uniformly distributed random numbers from the interval $(0, 1)$. They contribute to the stochastic nature of the algorithm.

1.10.3 Clamping at maximum values:

A particle’s velocity may be optionally clamped to a maximum value $v_{max} = [v^1_{max}, v^2_{max}, \ldots, v^D_{max}]$.If the velocity of the particle, at d-th dimension v^d exceeds v^d_{max} , specified by the user, then that value can be assigned to the nearest boundary value, or the value can be generated afresh within the limits. The clamping of the values of the velocities is generally done via:

$$v_d = sign(v_d) * v_{max}$$

where,

$$sign(x) = \frac{|x|}{x}$$

1.10.4 Reinitialization of particles:

A similar kind of treatment is done in case the particles get out of the bound that provided by the user i.e. they leave the search space. In that case, the particles who leave , are reinitialized afresh. By reinitialization, we mean, they are again assigned positions and velocities at random and made to fly the search space from scratch.

Chapter 2

Related Work

Particle swarm optimization is widely known due to its wide application in optimization domain. Mathematical analysis of the dynamics of PSO has attracted a good deal of research interest over the last decade and some comprehensive surveys on the research on and with PSO can be found in [19][25][24][5] The research efforts began with treatments relying on a number of simplifying, e.g. a single and isolated particle, stagnation (i.e. the local and globally best positions does not change with time), and deterministic conditions (no random number used for velocity updation).However, gradually more and more complexities were added to the particle dynamics and the analyses were undertaken by using more sophisticated mathematical tools.

2.1 Initial Work

The first analytical model of a single particle moving deterministically towards fixed attractors was put forward by Ozcan and Mohan in [26]. In [27], the work was extended to cover multiple, multi-dimensional particles and the authors demonstrated that a particle following the deterministic PSO dynamics describes a sinusoidal trajectory with random amplitude and frequency.

The first stability analysis of the simplified PSO dynamics was undertaken by Clerc and Kennedy [10], who considered similar simplifications as Ozcan and Mohan did. Following this model, the state-vector of a particle consists of its velocity and position and can be determined by finding the eigenvalues and eigenvectors of the state transition matrix. Clerc and Kennedy studied stable and limit cyclic behavior of the dynamics for the settings of appropriate values to its parameters and also added a new component, called 'constriction factor' to the velocity updation formula for PSO, taking a que from their analysis.

F van den Bergh undertook an independent theoretical analysis of the particle swarm dynamics and studied the particle trajectories under different conditions in his Ph. D thesis [32], published in the same year (2002). Blackwell [3] extended Clerc and kennedy's model, where the particles were allowed to interact and change their personal bests, and showed that under restrictive assumptions, the spread of the swarm in space decreases exponentially. Brandstätter and Baumgartner [4] related the PSO model of Clerc and Kennedy [10] with a damped mass-spring oscillator, making it possible to rewrite the model using the notions of damping factor and natural frequency of vibration. Trelea [31] undertook the analysis of a four-parameter family of particle models and identified regions in the parameter space where the model exhibits qualitatively distinct behaviors (stability, harmonic oscillations, or Brownian motion). Emara and Fattah [11] performed a similar stability analysis of the PSO dynamics in continuous time domain. Campana et al. [6], [7] modeled particles as a discrete, linear, and stationary dynamical system with no randomness and were able to formally express the free and forced responses.

2.2 Work by Kadirkamathanan:

Kadirkamanathan et al. [17] undertook a more generalized stability analysis of particle dynamics in presence of stochasticity, based on Lyapunov stability theorems. The authors derived stability conditions by regarding the PSO equations as a time-invariant linear second-order dynamic model, but did not make any general recommendations concerning the parametric set-up of PSO. Based on the analysis of [17], in [29] Samal et al. investigated an optimal parameter setting for PSO based on a closed loop stability analysis. The authors conducted experiments with 21 numerical benchmarks and concluded that the parameter settings ω= 0.6,a= 0.103 and b = 2.897 provide the best results. Jiang et al. [16] undertook a convergence analysis of PSO in presence of randomness, based on the stochastic process theory. The authors showed that each particle will converge to the global best position found by the swarm for a certain range of parameter settings; however, they did not establish that the global best position will actually correspond to the optimum of the optimization problem. Recently Poli [28] analyzed the characteristics of a PSO sampling distribution and explained how it changes over the generations, in presence of stochasticity and during stagnation.

2.3 Work by Martínez and Gonzalo

Fernandez-Martínez and García Gonzalo [23] derived a general continuous form of the PSO-dynamics(the 'GPSO' model), where they used an infinitesimal time interval Δt instead of the unity time interval originally used for the velocity and position update of PSO. They analyzed the deterministic and stochastic stability regions and their respective asymptotic velocities of convergence as a function of the time step Δt and the GPSO parameters. Recently the same authors undertook a very realistic stochastic analysis of the linear continuous and the generalized PSO models [12] using the theory of stochastic difference and differential equations for the most general case: transient solutions with a stochastic centre of attraction. Based on a statistical interpretation of the PSO model, Chen and Jiang [9] investigated the effect of particle interaction by considering the social-only model and derived the upper and lower bounds of the expected particle norm. Sudholt and Witt [30] theoretically investigated the run-time of a binary PSO for optimizing pseudo-Boolean functions. Gao and Xu proposed a new particle swarm optimizer with its global convergence behavior in 2011 [13]. Chen and Li also proposed a new particle swarm optimizer called PSO-CREV in 2007 [8], they mathematically tried to prove the exploration and convergence of their algorithm.

2.4 Contribution of this thesis:

Current work analyses and generalizes the PSO dynamics in continuous time domain following the thread of Fernandez-Martínez and García Gonzalo [23] A basic dynamical model is obtained by means of linearization process around a critical point.The notion of critical points and their nature of stability are also introduced. The conditions for stability of system , depending on the parametric values are obtained and justified, and thus the stable zone in the parametric space is calculated. Later on a detailed expression corresponding to the solution of the dynamics is obtained. Furthermore, it is shown that the dynamics shows similar nature irrespective of the simplifying assumptions made. At the end, concepts of Lyapunov stability is introduced in case of PSO dynamics, and it is shown , it is possible to calculate the settling time of a PSO dynamics , given initial conditions and proper Laypunov functions.

The thesis is organized as follows:
in Chapter 3, the PSO algorithm is elaborately explained, with help of simple examples. In Chapter 4 , the dynamics is obtained along the line of Martínez and Gonzalo [23] and linearized to study the nature of the system near the equilibrium points. Chapter 5 consists of the simulation results which prove the claims stated. Chapter 6 mathematically obtains an expression of the whole PSO system , without any assumptions, and further Chapter 7, talks about Lyapunov stability and its application in case of PSO dynamics and finally Chapter 8 shows possible options of future works and thus concludes the thesis.

Chapter 3

PSO Algorithm

As the name suggests , this kind of optimization technique works with the help of a swarm of ‘particles’. Given an optimization problem and the range of its solutions , a number of candidate solutions are chosen randomly from the range specified. Associated with each of the solutions, are a position and a velocity. The ‘position’ indicates where the particle does reside on the search space i.e. the range specified previously, and the ‘velocity’ dictates where the particle is going to move in the next iteration.

As an example, suppose we need to find out the minimum value of the quadratic function, x^2, where the search space is all over the real set. We randomly choose 10 candidate solutions. Suppose they are as follows:

$x =$	-5	-0.5	7	5	19	-43	32	1	0.005	0.45
$f(x) = x^2$	25	0.25	49	25	271	1849	1024	1	$5 * 10^{-4}$	0.2025

We call the ensemble of these candidate solutions a swarm and call each one of them a particle. As we can clearly see, the least value of x^2 occurs when $x = 0.005$. So, currently the value of x giving the best value of the objective function , is $x = 0.005$. Thus, we call this value of $f(x)$, the global best value and the corresponding value of x , the global best position.

Now we are not sure that whether the best position seen so far , is optimal or not, i.e. in this case whether 0.005 is actually giving the best solution for the optimization problem taken into consideration. Thus we try to randomly modify these solutions, and check if some modified solutions gives better results or not. So, we define something called 'velocity' and add that up with the candidate solutions. Each of the candidate solutions are going to have one velocity for themselves.

This velocity is determined by the global best position and the local best position obtained until the previous iteration, and determines how much the position is going to change in case of the current iteration from the previous ne. Hence the name velocity.

Velocity in current iteration is computed by adding up velocity in previous iteration, difference between local best and position in previous iteration, and difference between global best and position in previous iteration , all scaled up by some parameters.

Mathematically,

$$V(t+1) = \omega.V(t) + a.rand_1.(l(t) - x(t)) + b.rand_2.(g(t) - x(t)) \tag{3.1}$$

So, the position in the next iteration becomes,

$$x(t+1) = x(t) + V(t+1) \tag{3.2}$$

So, in the current case , the particles are all 1-dimensional and holding only 1 real value. The choice of the scaling up parameters, e.g.a, b, ω, are choices of the user. However, not every set of values of the parameters will end up producing satisfactory results for the algorithm. This problem is discussed in the current work.

Having a definite set of parameters in mind , we can tackle the problem of finding min in case of $f(x) = x^2$, starting with the initial values of the particles already stated. Let's choose $a = 0.5$, $b = 0.7$ & $\omega = 0.8$.

Now , the next job is to assign a set of values of velocities to the particles and let's assume , the velocities , randomly assigned within a random boundary of $(-5, 5)$ are as follows:

$x =$	-5	-0.5	7	5	19	-43	32	1	0.005	0.45
$f(x) = x^2$	25	0.25	49	25	271	1849	1024	1	$5 * 10^{-4}$	0.2025
$v =$	3.30	0.85	0.49	4.17	-2.14	2.57	2.53	-1.19	0.67	-4.24

According to equations 3.1 & 3.2 stated above , the next values the particles are going to take is, putting the values of parameters,

$$x(t+1) = x(t) + \omega.V(t) + a.rand_1.(l(t) - x(t)) + b.rand_2.(g(t) - x(t))$$

$$\Rightarrow x(t+1) = x(t) + 0.8 * V(t) + 0.5 * rand_1.(l(t) - x(t)) + 0.7.rand_2.(g(t) - x(t))$$

So, the table now becomes

$x =$	-5	-0.5	7	5	19	-43	32	1	0.005	0.45
$f(x) = x^2$	25	0.25	49	25	271	1849	1024	1	$5 * 10^{-4}$	0.2025
$v(t) =$	3.30	0.85	0.49	4.17	-2.14	2.57	2.53	-1.19	0.67	-4.24
$x(t+1) =$	0.91	0.51	2.81	5.07	4.86	-12.82	13.10	-0.60	0.54	-3.23

It's seen that all the particles now have values coming significantly closer to 0.005, and in successive iterations, it will come more so. But , the particle with value 0.005 itself has moved to 0.54. The reason behind this is the *localbest* and *globalbest* attractors are compelling the particles to move in the direction towards the attractors in action. The initial position of the initial global best ,although dismantled initially, will be restored in the subsequent iterations. This is shown in the figures in the next page. These figures graphically show how the PSO algorithm converges to find the best solution. Although initially all particles are scattered in the search space, eventually they come to an extremum, where they all have the same local best, global best and position, which is the solution to the problem.

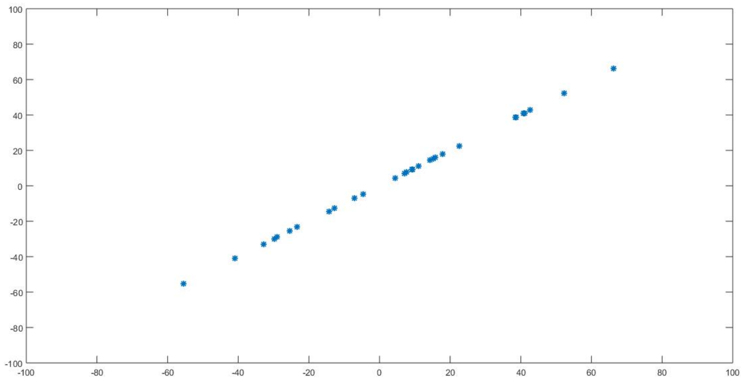


Figure 3.1: Illustration of PSO , Iteration 5: $a = 0.15, b = 0.25, \omega = -0.05$, number of particles = 30 , dimension = 1

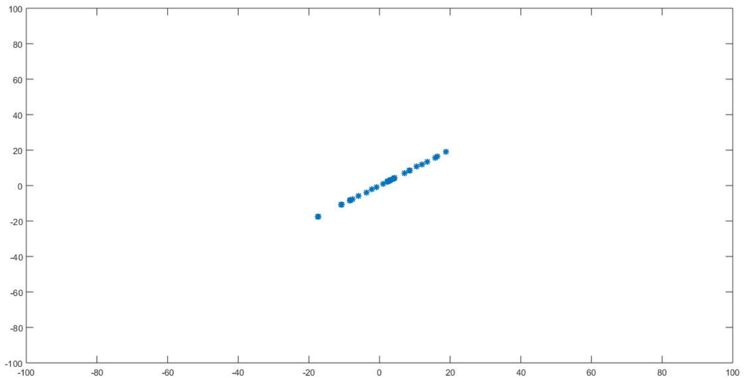


Figure 3.2: Illustration of PSO , Iteration 20: $a = 0.15, b = 0.25, \omega = -0.05$, number of particles = 30 , dimension = 1

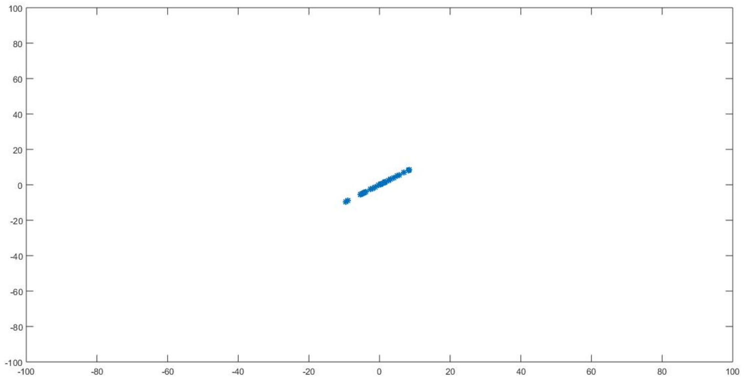


Figure 3.3: Illustration of PSO , Iteration 30: $a = 0.15, b = 0.25, \omega = -0.05$, number of particles = 30 , dimension = 1

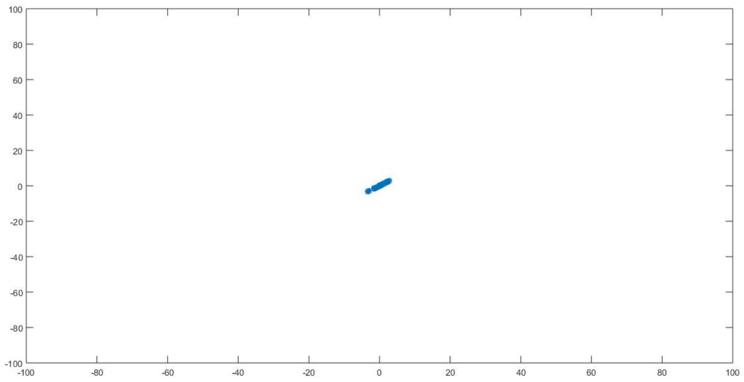


Figure 3.4: Illustration of PSO , Iteration 45: $a = 0.15, b = 0.25, \omega = -0.05$, number of particles = 30 , dimension = 1

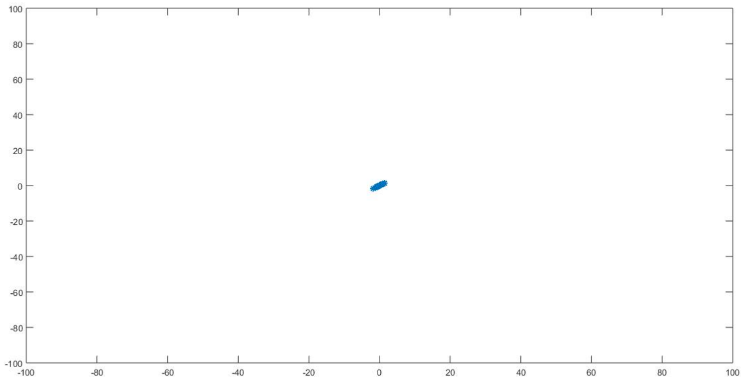


Figure 3.5: Illustration of PSO , Iteration 55: $a = 0.15, b = 0.25, \omega = -0.05$, number of particles = 30 , dimension = 1

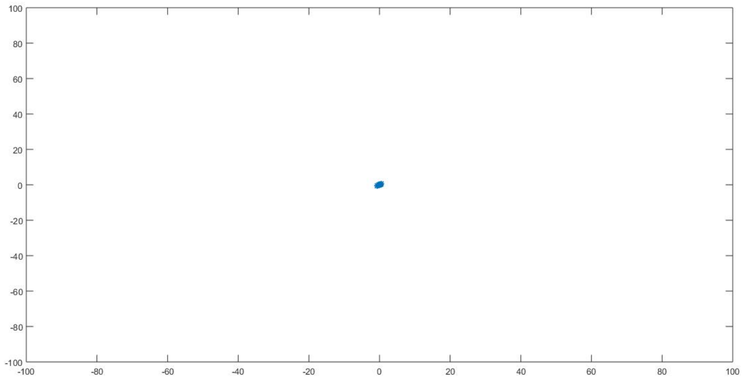


Figure 3.6: Illustration of PSO , Iteration 70: $a = 0.15, b = 0.25, \omega = -0.05$, number of particles = 30 , dimension = 1

Chapter 4

Linearization and Equilibrium Points of PSO

4.1 Analysis

From the idea of GPSO by Gonzalo and Martinez, let's start with the incremental PSO equation i.e.

$$\begin{aligned} v_i(t+\Delta t) &= (1-(1-\omega).\Delta t).v_i(t) + a.rand_1.\Delta t.(l_i(t) - x_i(t) \\ &\quad + b.rand_2.\Delta t.(g(t) - x_i(t) \\ x_i(t+\Delta t) &= x_i(t) + v_i(t+\Delta t)\Delta t \end{aligned} \tag{4.1}$$

where,

- $x_i(t)$ = position of ith particle at time instant t
- $v_i(t)$ = velocity of ith particle at time instant t
- $l_i(t)$ = local best OR particle best of ith particle at time instant t
- $g(t)$ = global best of the swarm at time instant t
- Δt = any time interval
- $rand_1$ & $rand_2$ = two uniformly generated random numbers in the range $(0, 1)$
- a, b, w = real constants

From equation 4.1,

$$\begin{aligned} v_i(t+\Delta t) &= \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} \\ \frac{v_i(t+\Delta t) - v_i(t)}{\Delta t} &= -(1-\omega).v_i(t) + a.rand_1.(l_i(t) - x_i(t)) \\ &+ b.rand_2.(g(t) - x_i(t)) \end{aligned} \tag{4.2}$$

Taking limits as $\lim\limits_{\Delta t \to 0}$, from equation 4.2 : (discrete domain to continuous domain)

$$\begin{aligned} v_i(t) &= \frac{d}{dt}(x_i(t)) = \dot{x}_i(t) \\ \dot{v_i}(t) &= -(1-\omega).v_i(t) + a.rand_1.(l_i(t) - x_i(t)) \qquad + b.rand_2.(g(t) - x_i(t)) \end{aligned} \tag{4.3}$$

By definition, a stochastic process or random process is a collection of random variables, which represents the evolution of some system of random values over time.

Thus, $x_i(t)$ & $v_i(t)$, being random variables , and initialized randomly , both constitute stochastic processes. So the **expectation** or mean of these processes should also satisfy the equations stated above. So, applying the expectation operator on the equation 4.3 , we get,

$$\begin{aligned} & E(\dot{x_i}(t)) = E(v_i(t)) \\ \Rightarrow & \frac{d}{dt}[E(x_i(t))] = E(v_i(t)) \\ \Rightarrow & \frac{d}{dt}\mu_x(t) = \mu_v(t) \\ \Rightarrow & \dot{\mu_x}(t) = \mu_v(t) \end{aligned} \tag{4.4}$$

where,

$$
\begin{aligned}
E(x_i(t)) &= \mu_x(t) \\
E(v_i(t)) &= \mu_v(t)
\end{aligned}
$$

And, also by linearity & multiplicity property of expectation operator, from equation 4.3

$$
\begin{aligned}
E(\dot{v_i(t)}) &= -(1-\omega).E(v_i(t)) + a.E(rand_1).E(l_i(t) - x_i(t)) \\
&+ b.E(rand_2).E((g(t) - x_i(t)) \\
&= -(1-\omega).E(v_i(t)) + a.\frac{1}{2}.E(l_i(t) - x_i(t)) + b.\frac{1}{2}.E((g(t) - x_i(t)) \\
&= -(1-\omega).E(v_i(t)) + a.\frac{1}{2}.(E(l_i(t)) - E(x_i(t)) \\
&+ b.\frac{1}{2}.E((g(t)) - E(x_i(t)) \\
&= -(1-\omega).E(v_i(t)) + a.\frac{1}{2}.(L(t)) - E(x_i(t)) \\
&+ b.\frac{1}{2}.(g(t) - E(x_i(t))
\end{aligned}
$$

Since, this equation 4.1 is all about one single i_{th} iteration, the expectation $E(x_i(t))$ is taken over all particle bests in ith iteration.
Let's assume $E(l_i(t)) = L(t)$ and , since global best is same for all particles in one iteration, it's a constant. Thus, $E(g(t)) = g(t)$. Thus,

$$
\begin{aligned}
\dot{\mu_v(t)} &= -(1-\omega).\mu_v(t) + a.\frac{1}{2}.(L(t) - \mu_x(t)) + b.\frac{1}{2}.(g(t) - \mu_x(t)) \\
&= -(1-\omega).\mu_v(t) + \frac{a+b}{2}.\mu_x(t) + \frac{1}{2}.(a.L(t) + b.g(t))
\end{aligned}
\tag{4.5}
$$

. Writing these equations in matrix form, from equation 4.4 and equation 4.5

$$\begin{bmatrix}\dot{\mu_x(t)}\\ \dot{\mu_v(t)}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{a+b}{2} & -(1-\omega)\end{bmatrix}\begin{bmatrix}\mu_x(t)\\ \mu_v(t)\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}\cdot\begin{bmatrix}\frac{a.L(t)+b.g(t)}{2}\end{bmatrix}$$

$$\Rightarrow \dot{y}(t) = \begin{bmatrix}0 & 1\\ -\frac{a+b}{2} & -(1-\omega)\end{bmatrix} y(t) + \begin{bmatrix}0\\1\end{bmatrix}.r(t) \tag{4.6}$$

where,

$$y(t) = \begin{bmatrix}\mu_x(t)\\ \mu_v(t)\end{bmatrix}$$

$$r(t) = \frac{(a.L(t)+b.g(t)}{2}$$

Thus, equation 4.6 gives us the final non linear dynamics of the PSO algorithm.

4.2 Linearization of the system

The generated system equation is not linear , as it has involvement of $r(t)$, in the system equation. According to Kadirkamanathan[17] , the system has

- no equilibrium point, when $l(t) \neq g(t)$
- one equilibrium point, at $(x, v) = (p, 0)$ when $l(t) = g(t) = p$

Thus, we get back to equation equation 4.5 and replace p for both $l(t)$ and $g(t)$.

$$\dot{\mu_v(t)} = -(1-\omega).\mu_v(t) + a.\frac{1}{2}.(p - \mu_x(t)) + b.\frac{1}{2}.(p - \mu_x(t)) \tag{4.7}$$

Along the lines of Kadirkamanathan [17], we can see, the equilibrium points are here also, $\mu_x = p, \mu_v = 0$. Also, we know that any equilibrium point can be considered as residing at the origin applying proper shift in the axes. To linearize, we expand the functions about this equilibrium point, using Taylor's series expansion.

Suppose, x_1 & x_2 are two state variables in the state space which have a value of p_1 & p_2 of equilibrium points. Thus,

$$\begin{aligned}\dot{x_1} &= f_1(p_1,p_2) + \frac{\partial f_1(x_1,x_2)}{\partial x_1}|_{x_1=p_1,x_2=p_2}(x_1-p_1)\\ &+ \frac{\partial f_1(x_1,x_2)}{\partial x_2}|_{x_1=p_1,x_2=p_2}(x_2-p_2) + H.O.T\\ \dot{x_2} &= f_2(p_1,p_2) + \frac{\partial f_2(x_1,x_2)}{\partial x_1}|_{x_1=p_1,x_2=p_2}(x_1-p_1)\\ &+ \frac{\partial f_2(x_1,x_2)}{\partial x_2}|_{x_1=p_1,x_2=p_2}(x_2-p_2) + H.O.T\end{aligned}$$

Where H.O.T = Higher Order Terms.

In current scenario,

$$f_1(p_1,p_2) = 0$$
$$f_2(p_1,p_2) = 0$$

as (p_1, p_2) is an equilibrium point. At equilibrium points in case of autonomous systems, $\dot{x} = 0 \Rightarrow f(x) = 0$, as $\dot{x} = f(x)$. Taking the partial derivatives at the equilibrium point, we obtain,

$$\begin{aligned}\frac{\partial f_1(x_1,x_2)}{\partial x_1}|_{p_1,p_2} &= \frac{\partial \mu_v(t)}{\partial \mu_x(t)}|_{p,0} = 0\\ \frac{\partial f_1(x_1,x_2)}{\partial x_2}|_{p_1,p_2} &= \frac{\partial \mu_v(t)}{\partial \mu_v(t)}|_{p,0} = 1\\ \frac{\partial f_2(x_1,x_2)}{\partial x_1}|_{p_1,p_2} &= \frac{\partial - \frac{a+b}{2}\mu_x(t) - (1-\omega).\mu_v(t)}{\partial \mu_x(t)}|_{p,0} = -\frac{a+b}{2}\\ \frac{\partial f_2(x_1,x_2)}{\partial x_2}|_{p_1,p_2} &= \frac{\partial - \frac{a+b}{2}\mu_x(t) - (1-\omega).\mu_v(t)}{\partial \mu_v(t)}|_{p,0} = -(\omega-1)\end{aligned}$$

Thus , neglecting higher order terms, we obtain the linearized equation in the vicinity of the equilibrium point $(p, 0)$ as:

$$\overset{\cdot}{\begin{bmatrix} \mu_x(t) \\ \mu_v(t) \end{bmatrix}} = \begin{bmatrix} 0 & 1 \\ -\frac{a+b}{2} & -(1-\omega) \end{bmatrix} \cdot \begin{bmatrix} \mu_x(t) \\ \mu_v(t) \end{bmatrix} \tag{4.8}$$

We now relax the condition imposed previously on the system that $a + b = 0$ and examine other equilibrium points of this linearized system. Assuming $y(t) = \begin{bmatrix} \mu_x(t) \\ \mu_v(t) \end{bmatrix}$, one can write the same equation as:

$$\dot{y}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{a+b}{2} & -(1-\omega) \end{bmatrix} .y(t) \tag{4.9}$$

Now, the solutions of autonomous equations of type $\dot{y}(t) = A.y(t)$ has the form

$$y(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$$

where λ_1 & λ_2 are eigenvalues of the matrix A and

$$\lambda = \frac{1}{2} \cdot \left(Tr(A) \pm \sqrt{(Tr(A))^2 - 4.Det(A)} \right),$$

where, Tr(A) = sum of elements in main diagonal

Depending on the values λ_is take, the phase portraits are of different shapes. This information can be used to determine the values of ω required for different kind of phase portraits.

4.3 Finding ranges of ω

4.3.1 Case 1: $\lambda_1 \neq \lambda_2 \neq 0$

Both the eigenvalues of A , are real , unequal and non zero. This happens iff $det(A) \neq 0$, because otherwise at least one λ would have been zero. Also, since both are real, the discriminant must be positive.

$\therefore (Tr(A))^2 > 4.Det(A)$.

Subcase 1: both λs are of same sign

Also, $det(A) > 0$, because otherwise $\sqrt{(Tr(A))^2 - 4.Det(A)} > Tr(A)$ i.e. both eigenvalues wouldn't be of the same sign. Thus, $a + b > 0$, $\because det(A) = \frac{a+b}{2}$.

Now,

$$
\begin{aligned}
&Tr(A)^2 > 4.det(A)\\
&\Rightarrow (1-\omega)^2 > 2(a+b)\\
&\Rightarrow \omega^2 - 2.\omega + 1 > 2(a+b)\\
&\Rightarrow \omega^2 - 2.\omega + (1 - 2(a+b)) > 0
\end{aligned}
$$

for the expression on the LHS to be always positive, one of the following must be true:

1. $\omega > \omega_1$ & $\omega > \omega_2$

2. $\omega < \omega_1$ & $\omega < \omega_2$

where , ω_1 & ω_2 are roots of the equation,

$$\omega^2 - 2.\omega + (1 - 2(a+b)) = 0$$

and are equal to

$$\omega_1 = 1 + \sqrt{2(a+b)}, \qquad \omega_2 = 1 - \sqrt{2(a+b)}$$

- for condition 1 : $\omega > 1 + \sqrt{2(a+b)}, \qquad \& \qquad \omega > 1 - \sqrt{2(a+b)}$, combining we get, $\omega > 1 + \sqrt{2(a+b)}$, as $(a+b) > 0$

- for condition 2 : $\omega < 1 + \sqrt{2(a+b)}, \qquad \& \qquad \omega < 1 - \sqrt{2(a+b)}$, combining we get, $\omega < 1 - \sqrt{2(a+b)}$, as $(a+b) > 0$

Subsubcase 1: $\lambda_1 > \lambda_2 > 0$

If $\lambda_1 > \lambda_2 > 0$, $\lambda_1 + \lambda_2 > 0$, $\therefore Tr(A) > 0$. Thus, $-(1-\omega) > 0$, $\because Tr(A) > 0$. $\therefore \omega > 1$.

Thus, additional conditions for both positive eigenvalues are $(a + b) > 0$ and $w > 1$.

$$\begin{aligned} &\text{Condition 1}: \omega > 1 + \sqrt{2(a+b)} &&\& \omega > 1 &&\Rightarrow \omega > 1 + \sqrt{2(a+b)} \\ &\text{Condition 2}: \omega < 1 - \sqrt{2(a+b)} &&\& \omega > 1 &&\Rightarrow impossible \end{aligned}$$

An unstable node is obtained in case of both positive eigenvalues. So, for obtaining an **unstable node**, $\omega \in (1 + \sqrt{2(a+b)}, \infty)$.

Subsubcase 1: $\lambda_2 < \lambda_1 < 0$

If $\lambda_2 < \lambda_1 < 0$, $\lambda_1 + \lambda_2 < 0$, $\therefore Tr(A) < 0$. Thus, $-(1 - \omega) < 0$, $\because Tr(A) < 0$. $\therefore \omega < 1$.

Thus, additional conditions for both negative eigenvalues are $(a + b) > 0$ and $\omega < 1$.

$$\begin{aligned} &\text{Condition 1}: \omega > 1 + \sqrt{2(a+b)} &&\& \omega < 1 &&\Rightarrow impossible \\ &\text{Condition 2}: \omega < 1 - \sqrt{2(a+b)} &&\& \omega < 1 &&\Rightarrow \omega < 1 - \sqrt{2(a+b)} \end{aligned}$$

A stable node is obtained in case of both negative eigenvalues. So, for obtaining a **stable node**, $\omega \in (-\infty, 1 - \sqrt{2(a+b)})$.

Subcase 2: both λs are of different sign

$det(A) < 0$ because otherwise both eigenvalues will have same sign. Thus, $a+b < 0$, as $det(A) = a + b$. A saddle point is obtained in case of one negative and one positive eigenvalues. So , to obtain a saddle point, just the condition $a + b < 0$ is sufficient. Interestingly, this does not depend on value of ω.

4.3.2 Case 2: both eigenvalues complex

Obviously, the discriminant will be less than zero in this case. $\therefore (Tr(A))^2 < 4.Det(A)$.By similar arguments,it leads to:

$$\omega^2 - 2.\omega + (1 - 2(a+b)) < 0$$

Now, for the expression on the LHS to be always negative, one of the following must be true:

1. $\omega < \omega_1$ & $\omega > \omega_2$
2. $\omega > \omega_1$ & $\omega < \omega_2$

where , ω_1 & ω_2 are as stated previously.

- for condition 1 :
  $\omega < 1 + \sqrt{2(a+b)}, \qquad \& \qquad \omega > 1 - \sqrt{2(a+b)},$
  combining we get, $\omega \in (1 - \sqrt{2(a+b)}, 1 + \sqrt{2(a+b)})$
- for condition 2 :
  $\omega > 1 + \sqrt{2(a+b)}, \qquad \& \qquad \omega < 1 - \sqrt{2(a+b)},$
  which is, an impossible condition.

Subcase 1: $\Re(\lambda) > 0$

If $\Re(\lambda) > 0$, $Tr(A) > 0$. As $Tr(A) = \Re(\lambda)$. Thus, additional conditions for both complex eigenvalues with positive real part , are $\omega > 1$.

$$\text{Condition 1 : } \omega \in (1 - \sqrt{2(a+b)}, 1 + \sqrt{2(a+b)}) \qquad \& \omega > 1$$
$$\Rightarrow \omega \in (1, 1 + \sqrt{2(a+b)})$$

An unstable spiral is obtained in case of both complex eigenvalues with positive real part. So, for obtaining an **unstable spiral**, $\omega \in (1, 1 + \sqrt{2(a+b)})$.

Subcase 2: $\Re(\lambda) < 0$

If $\Re(\lambda) < 0$, $Tr(A) < 0$. As $Tr(A) = \Re(\lambda)$. Thus, additional conditions for both complex eigenvalues with negative real part , are $\omega < 1$.

$$\begin{aligned} \text{Condition 1}: \omega \in (1 - \sqrt{2(a+b)}, 1 + \sqrt{2(a+b)}) \qquad & \& \omega < 1 \\ \Rightarrow \omega \in (1 - \sqrt{2(a+b)}, 1) \end{aligned}$$

A stable spiral is obtained in case of both complex eigenvalues with negative real part. So, for obtaining a **stable spiral**, $\omega \in (1 - \sqrt{2(a+b)}, 1)$.

Subcase 3: $\Re(\lambda) = 0$

If $\Re(\lambda) = 0$, $Tr(A) = 0$. As $Tr(A) = \Re(\lambda)$. Thus, additional conditions for both complex eigenvalues with no real part , are $\omega = 1$.

$$\begin{aligned} \text{Condition 1}: \omega \in (1 - \sqrt{2(a+b)}, 1 + \sqrt{2(a+b)}) \qquad & \& \omega = 1 \\ \Rightarrow \omega = 1 \end{aligned}$$

A center is obtained in case of both complex eigenvalues with negative real part. It is seen to have limit cyclic behavior in this case.So, for obtaining a **limit cycle**, $\omega = 1$.

4.3.3 Case 3: Non Zero multiple eigenvalue

Both eigenvalues are equal iff, the discriminant is zero. $\therefore (Tr(A))^2 = det(A)$. By similar arguments, $\omega^2 - 2.\omega + (1 - 2(a+b)) = 0$. $\therefore \omega = 1 - \sqrt{2(a+b)}, 1 + \sqrt{2(a+b)}$.

Subcase 1: $\lambda > 0$

Again, $2\lambda > 0 \Rightarrow Tr(A) > 0$. $\because Tr(A) = \lambda_1 + \lambda_2$. Thus, $Tr(A) = -(1-\omega) > 0 \Rightarrow \omega > 1$. Thus, $\omega = 1 + \sqrt{2(a+b)}$.

The equilibrium point is called an unstable node in this case also.

Subcase 2: $\lambda < 0$

Again, $2\lambda < 0 \Rightarrow Tr(A) < 0$. $\because Tr(A) = \lambda_1 + \lambda_2$. Thus, $Tr(A) = -(1-\omega) < 0 \Rightarrow \omega < 1$. Thus, $\omega = 1 - \sqrt{2(a+b)}$.

The equilibrium point is called a stable node in this case also.

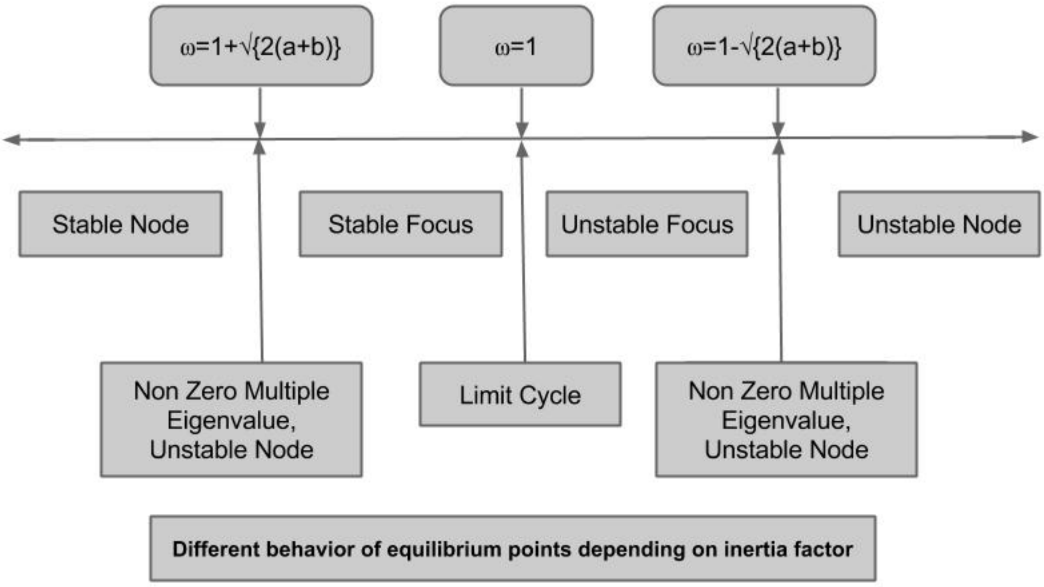
4.3.4 Table of possible cases

The eigenvalues can be of the following nature:

Case	(λ)	$(a+b)$	ω	Name, Stability	Stability
1	both real, unequal,+ve	> 0	$(1+\sqrt{2(a+b)}, \infty)$	Node	Unstable
2	both real, unequal, -ve	> 0	$(-\infty, 1-\sqrt{2(a+b)})$	Node	Stable
3,4	both real, Opposite sign	< 0	–	Saddle	Unstable
5	both complex, $Re(\lambda) \neq 0$	> 0	$(1-\sqrt{2(a+b)}, 1)$	Focus	Stable
6	–	> 0	1	Limit Cycle	Stable
7	both complex, $Re(\lambda) \neq 0$	> 0	$(1, 1+\sqrt{2(a+b)})$	Focus	Unstable
8	Non Zero multiple eig val	> 0	$1+\sqrt{2(a+b)}$	also called node	Unstable
9	Non Zero multiple eig val	> 0	$1-\sqrt{2(a+b)}$	also called node	Stable

4.3.5 Summarization:

Summarizing, the value of ω dictates the nature of the $fixedpoints$, which in turn, depends on the value of the PSO parameters shown in figure next.



ω=1+√{2(a+b)}
ω=1
ω=1-√{2(a+b)}
Stable Node
Stable Focus
Unstable Focus
Unstable Node
Non Zero Multiple Eigenvalue, Unstable Node
Limit Cycle
Non Zero Multiple Eigenvalue, Unstable Node
Different behavior of equilibrium points depending on inertia factor

Figure 4.1: variation of ω for different fixed points

Chapter 5

Simulations

The simulations are obtained using MatLab for a number of different functions. The cases for multidimensional sphere function are as follows:

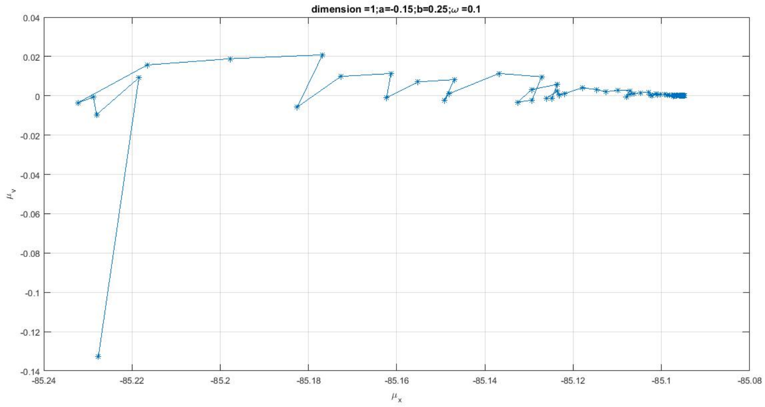


Figure 5.1: Stable Node: $a = -0.15, b = 0.25, \omega = 0.1$

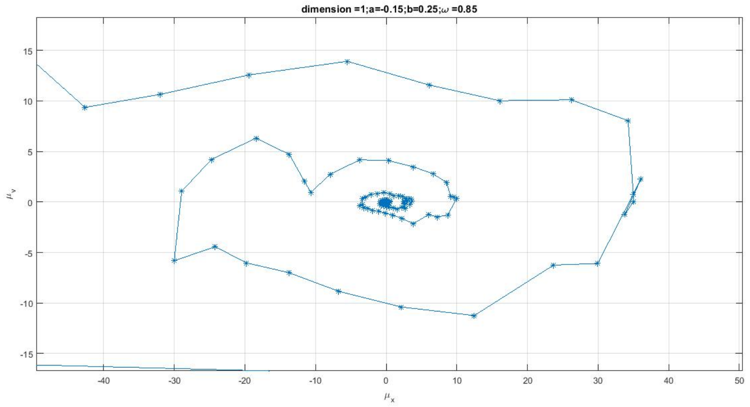


Figure 5.2: Stable Focus: $a = -0.15, b = 0.25, \omega = 0.85$

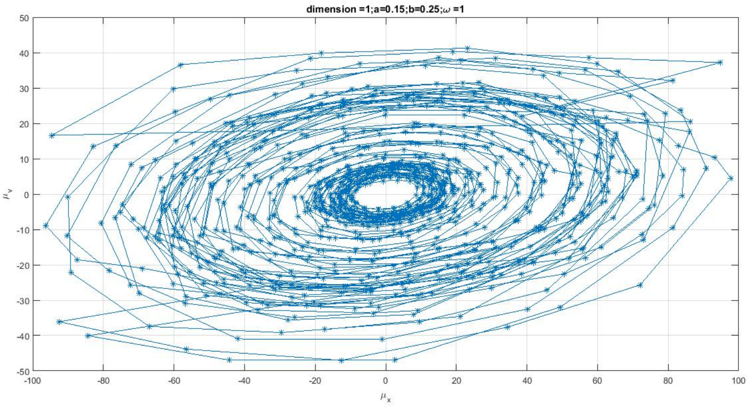


Figure 5.3: Stable Node: $a = 0.15, b = 0.25, \omega = 1$

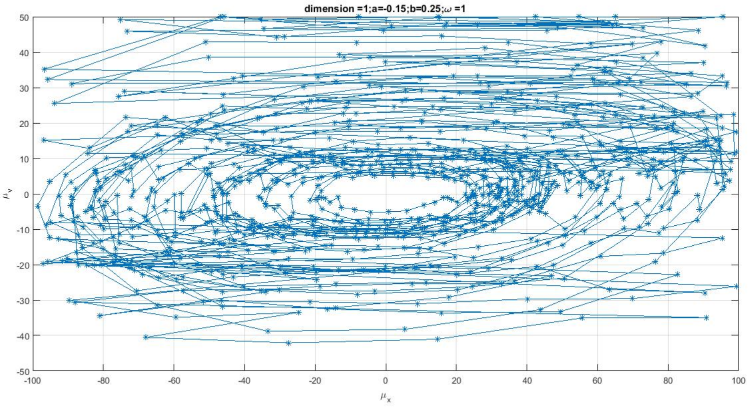


Figure 5.4: Limit Cycle: $a = -0.15, b = 0.25, \omega = 1$

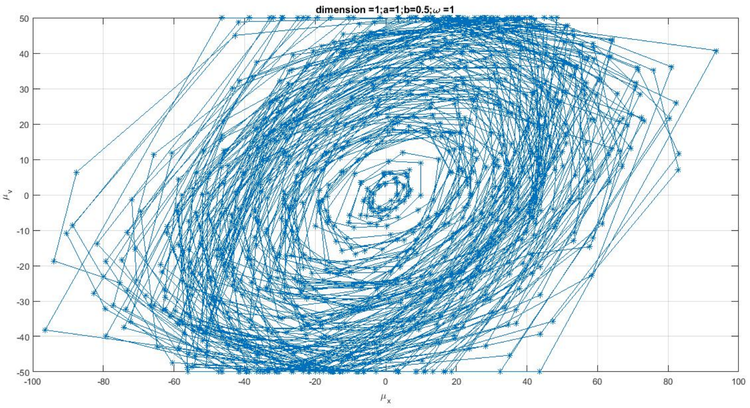


Figure 5.5: Limit Cycle: $a = 1, b = 0.5, \omega = 1$

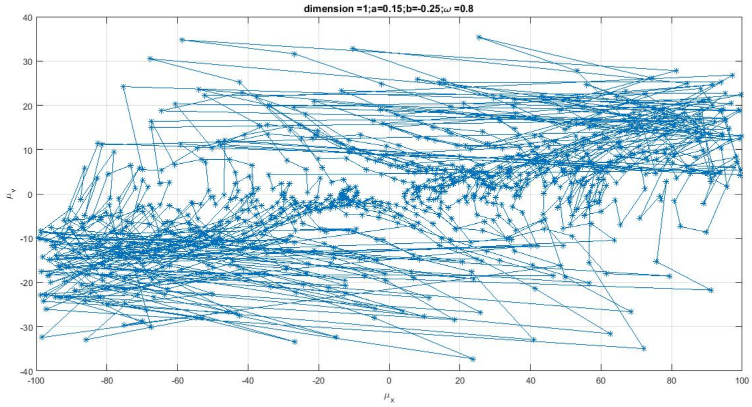


Figure 5.6: Saddle Point, only a is negative: $a = 0.15, b = -0.25, \omega = 0.8$

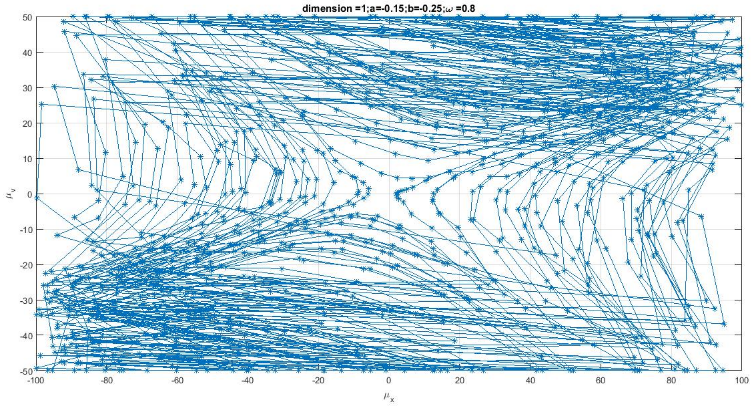


Figure 5.7: Saddle Point, both a and b are negative: $a = -0.15, b = -0.25, \omega = 0.8$

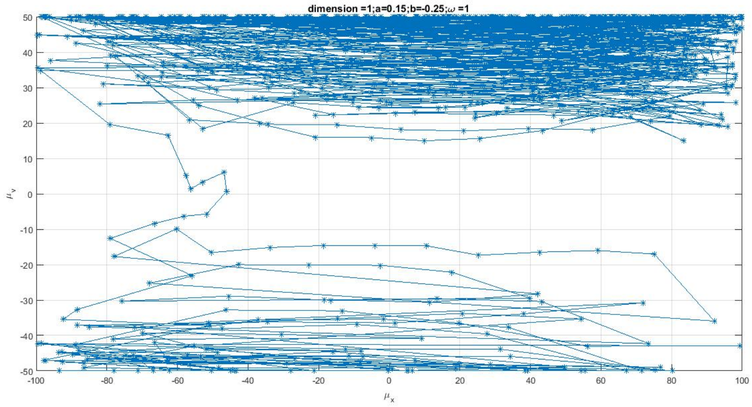


Figure 5.8: Saddle point: $a = 0.15, b = -0.25, \omega = 1$, shows that saddle point nature prevails over limit cycle

Chapter 6

Solving Non Linear System Dynamics

Now, the whole system is a non homogeneous linear system, of the following form:

$$\dot{x} = F(x) = A.x + h(t)$$

where, $h(t)$ is a function of time. The mapping F is said to be affine mapping. Applying the matrix equivalent of 'Integrating factor method' , [2]

$$\begin{aligned}
&\dot{x} = A.x + h(t) \\
&\Rightarrow e^{-At}.\dot{x} = e^{-At}.Ax + e^{-At}.h(t) \\
&\Rightarrow \frac{d}{dt}(x.e^{-At}) = e^{-At}.h(t)
\end{aligned}$$

Integrating from t_0 to t,

$$e^{-At}.x(t) - e^{-At_0}.x(t_0) = \int_{t_0}^{t} e^{-As}.h(s).ds$$

$\therefore$

$$\Rightarrow x(t) = \underbrace{e^{A(t-t_0)}.x(t_0)}_{\text{Complimentary Function}} + \underbrace{e^{At}.\int_{t_0}^{t} e^{-As}.h(s).ds}_{\text{Particular Integral}} \tag{6.1}$$

$\therefore$ for the current case,

$$\begin{aligned} y_{CF}(t) &= e^{A(t-t_0)}.y(t_0) \\ &= e^{A.t}.y(0) \qquad \text{, since } t_0 = 0; \\ &= \left[e^{\lambda_1.t}.Q_1 + e^{\lambda_2.t}.Q_2 \right] \cdot \begin{bmatrix} \mu_x(0) \\ \mu_v(0) \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} Q_1 &= \frac{A - \lambda_2.I}{\lambda_1 - \lambda_2} \\ Q_1 &= \frac{A - \lambda_2.I}{\lambda_1 - \lambda_2} \end{aligned}$$

λ_1 & λ_2 being eigenvalues of matrix A, $\lambda_1 \neq \lambda_2$. I being identity matrix of required order.

$$y_{PI}(t) = e^{At} \int_{t_0}^{t} e^{-As} h(s) ds$$

$$\begin{aligned} & e^{At} \\ &= \frac{e^{\lambda_1 t}}{\lambda_1 - \lambda_2}. \begin{bmatrix} -\lambda_2 & 1 \\ -\frac{a+b}{2} & -(1-\omega) - \lambda_2 \end{bmatrix} + \frac{e^{\lambda_2 t}}{\lambda_1 - \lambda_2}. \begin{bmatrix} -\lambda_1 & 1 \\ -\frac{a+b}{2} & -(1-\omega) - \lambda_1 \end{bmatrix} \\ &= \frac{1}{\lambda_1 - \lambda_2}. \begin{bmatrix} (-\lambda_2.e^{\lambda_1 t} - \lambda_1.e^{\lambda_2 t}) & (e^{\lambda_1 t} + e^{\lambda_2 t}) \\ -(e^{\lambda_1 t} + e^{\lambda_2 t})(\frac{a+b}{2}) & (\omega - 1 - \lambda_2).e^{\lambda_1 t} + (\omega - 1 - \lambda_1).e^{\lambda_2 t}) \end{bmatrix} \end{aligned}$$

Replacing t by $-s$,

$$e^{-As}$$

$$= \frac{1}{\lambda_1 - \lambda_2}. \begin{bmatrix} (-\lambda_2.e^{-\lambda_1 s} - \lambda_1.e^{-\lambda_2 s}) & (e^{-\lambda_1 s} + e^{-\lambda_2 s}) \\ -(e^{-\lambda_1 s} + e^{-\lambda_2 s})(\frac{a+b}{2}) & (\omega - 1 - \lambda_2).e^{-\lambda_1 s} + (\omega - 1 - \lambda_1).e^{-\lambda_2 s}) \end{bmatrix}$$

Now,

$$h(s) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} .r(s) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \begin{bmatrix} \frac{a.L(s)+b.g(s)}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{a.L(s)+b.g(s)}{2} \end{bmatrix}$$

Thus,

$$\int_{t_0}^{t} e^{-As} h(s) ds$$

$$= \int_{t_0}^{t} .\frac{1}{\lambda_1 - \lambda_2}. \begin{bmatrix} (-\lambda_2.e^{-\lambda_1 s} - \lambda_1.e^{-\lambda_2 s}) & (e^{-\lambda_1 s} + e^{-\lambda_2 s}) \\ -(e^{-\lambda_1 s} + e^{-\lambda_2 s})(\frac{a+b}{2}) & ((\omega - 1 - \lambda_2).e^{-\lambda_1 s} + (\omega - 1 - \lambda_1).e^{-\lambda_2 s}) \end{bmatrix}.$$

$$\begin{bmatrix} 0 \\ \frac{a.L(s)+b.g(s)}{2} \end{bmatrix} .ds$$

$$= \int_{t_0}^{t} .\frac{1}{\lambda_1 - \lambda_2}. \begin{bmatrix} (e^{-\lambda_1 s} + e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2} \\ ((\omega - 1 - \lambda_2).e^{-\lambda_1 s} + (\omega - 1 - \lambda_1).e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2} \end{bmatrix} .ds$$

$$= \int_{t_0}^{t} .\frac{1}{\lambda_1 - \lambda_2}. \begin{bmatrix} (e^{-\lambda_1 s} + e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2} \\ ((\lambda_1).e^{-\lambda_1 s} + (\lambda_2).e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2} \end{bmatrix} .ds, \text{ Since } \lambda_1 + \lambda_2 = \omega - 1$$

$$\begin{aligned} y(t) &= e^{At}.y(t_0) + e^{A.t}. \int_{t_0}^{t} e^{-As}.h(s).ds \\ &= e^{At}(y(t_0) + \int_{t_0}^{t} e^{-As}.h(s).ds) \\ &= e^{At}(\begin{bmatrix} \mu_x(0) \\ \mu_v(0) \end{bmatrix} + \int_{t_0}^{t} e^{-As}.h(s).ds) \end{aligned}$$

Thus,

$$y(t) = e^{A.t}\begin{bmatrix} \mu_x(0) + \frac{1}{\lambda_1-\lambda_2}\int_{t_0}^{t}(e^{-\lambda_1 s} + e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2}.ds \\ \mu_v(0) + \frac{1}{\lambda_1-\lambda_2}\int_{t_0}^{t}(\lambda_1.e^{-\lambda_1 s} + \lambda_2.e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2}.ds \end{bmatrix}$$

Putting values of e^{At},

$$y(t) = \frac{1}{\lambda_1 - \lambda_2}.\begin{bmatrix} (-\lambda_2.e^{\lambda_1 t} - \lambda_1.e^{\lambda_2 t}) & (e^{\lambda_1 t} + e^{\lambda_2 t}) \\ -(e^{\lambda_1 t} + e^{\lambda_2 t})(\frac{a+b}{2}) & (\omega - 1 - \lambda_2).e^{\lambda_1 t} + (\omega - 1 - \lambda_1).e^{\lambda_2 t}) \end{bmatrix} *$$
$$\begin{bmatrix} \mu_x(0) + \frac{1}{\lambda_1-\lambda_2}\int_{t_0}^{t}(e^{-\lambda_1 s} + e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2}.ds \\ \mu_v(0) + \frac{1}{\lambda_1-\lambda_2}\int_{t_0}^{t}(\lambda_1.e^{-\lambda_1 s} + \lambda_2.e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2}.ds \end{bmatrix}$$

Now, since, $y(t) = \begin{bmatrix} \mu_x(t) \\ \mu_v(t) \end{bmatrix}$,

$$\begin{aligned}
\mu_x(t) &= \frac{1}{\lambda_1 - \lambda_2}(-\lambda_2.e^{\lambda_1 t} - \lambda_1.e^{\lambda_2 t}).\Big\{\mu_x(0) \\
&+ \frac{1}{\lambda_1 - \lambda_2}\int_{t_0}^{t}(e^{-\lambda_1 s} + e^{-\lambda_2 s})\frac{a.L(s) + b.g(s)}{2}.ds\Big\} \\
&+ \frac{1}{\lambda_1 - \lambda_2}(e^{\lambda_1 t} + e^{\lambda_2 t})\Big\{\mu_v(0) \\
&+ \frac{1}{\lambda_1 - \lambda_2}\int_{t_0}^{t}(\lambda_1.e^{-\lambda_1 s} + \lambda_2.e^{-\lambda_2 s})\frac{a.L(s) + b.g(s)}{2}.ds\Big\} \\
&= \frac{1}{\lambda_1 - \lambda_2}\Big\{(-\lambda_2.e^{\lambda_1 t} - \lambda_1.e^{\lambda_2 t}).\mu_x(0) + (e^{\lambda_1 t} + e^{\lambda_2 t}).\mu_v(0)\Big\} \\
&+ \frac{1}{(\lambda_1 - \lambda_2)^2}\Big\{(-\lambda_2.e^{\lambda_1 t} - \lambda_1.e^{\lambda_2 t})\int_{t_0}^{t}(e^{-\lambda_1 s} + e^{-\lambda_2 s})\frac{a.L(s) + b.g(s)}{2}.ds\Big\} \\
&+ \frac{1}{(\lambda_1 - \lambda_2)^2}\Big\{(e^{\lambda_1 t} + e^{\lambda_2 t})\int_{t_0}^{t}(\lambda_1.e^{-\lambda_1 s} + \lambda_2.e^{-\lambda_2 s})\frac{a.L(s) + b.g(s)}{2}.ds\Big\} \\
&= \frac{1}{\lambda_1 - \lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0) + \mu_v(0)) + e^{\lambda_2 t}(-\lambda_1\mu_x(0) + \mu_v(0))\Big\} \\
&+ \frac{1}{(\lambda_1 - \lambda_2)^2}\Big\{(-\lambda_2.e^{\lambda_1 t} - \lambda_1.e^{\lambda_2 t})\int_{t_0}^{t}(e^{-\lambda_1 s} + e^{-\lambda_2 s})\frac{a.L(s) + b.g(s)}{2}.ds\Big\} \\
&+ \frac{1}{(\lambda_1 - \lambda_2)^2}\Big\{(e^{\lambda_1 t} + e^{\lambda_2 t})\int_{t_0}^{t}(\lambda_1.e^{-\lambda_1 s} + \lambda_2.e^{-\lambda_2 s})\frac{a.L(s) + b.g(s)}{2}.ds\Big\}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\lambda_1-\lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0)+\mu_v(0))+e^{\lambda_2 t}(-\lambda_1\mu_x(0)+\mu_v(0))\Big\}\\
&+\frac{1}{(\lambda_1-\lambda_2)^2}\Big\{\int_{t_0}^{t}(-\lambda_2.e^{\lambda_1 t}-\lambda_1.e^{\lambda_2 t})(e^{-\lambda_1 s}+e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2}.ds\Big\}\\
&+\frac{1}{(\lambda_1-\lambda_2)^2}\Big\{\int_{t_0}^{t}(e^{\lambda_1 t}+e^{\lambda_2 t})(\lambda_1.e^{-\lambda_1 s}+\lambda_2.e^{-\lambda_2 s})\frac{a.L(s)+b.g(s)}{2}.ds\Big\}\\
&= \frac{1}{\lambda_1-\lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0)+\mu_v(0))+e^{\lambda_2 t}(-\lambda_1\mu_x(0)+\mu_v(0))\Big\}\\
&+\frac{1}{(\lambda_1-\lambda_2)^2}\int_{t_0}^{t}\Big[-\lambda_2 e^{\lambda_1 t-\lambda_1 s}-\lambda_1 e^{\lambda_2 t-\lambda_1 s}\\
&\qquad -\lambda_2 e^{\lambda_1 t-\lambda_2 s}-\lambda_1 e^{\lambda_2 t-\lambda_2 s}\Big]\frac{a.L(s)+b.g(s)}{2}ds\\
&+\frac{1}{(\lambda_1-\lambda_2)^2}\int_{t_0}^{t}\Big[\lambda_1 e^{\lambda_1 t-\lambda_1 s}+\lambda_1 e^{\lambda_2 t-\lambda_1 s}\\
&\qquad +\lambda_2 e^{\lambda_1 t-\lambda_2 s}+\lambda_2 e^{\lambda_2 t-\lambda_2 s}\Big]\frac{a.L(s)+b.g(s)}{2}ds\\
&= \frac{1}{\lambda_1-\lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0)+\mu_v(0))+e^{\lambda_2 t}(-\lambda_1\mu_x(0)+\mu_v(0))\Big\}\\
&+\frac{1}{(\lambda_1-\lambda_2)^2}\int_{t_0}^{t}\Big[-\lambda_2 e^{\lambda_1 t-\lambda_1 s}-\lambda_1 e^{\lambda_2 t-\lambda_2 s}\\
&\qquad +\lambda_1 e^{\lambda_1 t-\lambda_1 s}+\lambda_2 e^{\lambda_2 t-\lambda_2 s}\Big]\frac{a.L(s)+b.g(s)}{2}ds\\
&= \frac{1}{\lambda_1-\lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0)+\mu_v(0))+e^{\lambda_2 t}(-\lambda_1\mu_x(0)+\mu_v(0))\Big\}\\
&+\frac{1}{(\lambda_1-\lambda_2)^2}\int_{t_0}^{t}\Big[(\lambda_1-\lambda_2)e^{\lambda_1 t-\lambda_1 s}+(\lambda_2-\lambda_1)e^{\lambda_2 t-\lambda_2 s}\Big].\frac{a.L(s)+b.g(s)}{2}ds\\
&= \frac{1}{\lambda_1-\lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0)+\mu_v(0))+e^{\lambda_2 t}(-\lambda_1\mu_x(0)+\mu_v(0))\Big\}\\
&+\frac{1}{\lambda_1-\lambda_2}\int_{t_0}^{t}\Big[e^{\lambda_1 t-\lambda_1 s}-e^{\lambda_2 t-\lambda_2 s}\Big]\frac{a.L(s)+b.g(s)}{2}ds\\
&= \frac{1}{\lambda_1-\lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0)+\mu_v(0))+e^{\lambda_2 t}(-\lambda_1\mu_x(0)+\mu_v(0))\Big\}\\
&+\frac{1}{\lambda_1-\lambda_2}\int_{t_0}^{t}\Big[e^{\lambda_1(t-s)}-e^{\lambda_2(t-s)}\Big]\frac{a.L(s)+b.g(s)}{2}ds
\end{aligned}
$$

By similar arguments, we can show that,

$$
\begin{aligned}
\mu_v(t) &= \frac{\lambda_2(-\lambda_2\mu_x(0)+\mu_v(0))}{\lambda_1-\lambda_2}e^{\lambda_2 t}+\frac{\lambda_1(-\lambda_2\mu_x(0)-\mu_v(0))}{\lambda_1-\lambda_2}e^{\lambda_1 t}\\
&+\frac{1}{\lambda_1-\lambda_2}\int_{t_0}^{t}\Big[e^{\lambda_1(t-s)}-e^{\lambda_2(t-s)}\Big]\frac{a.L(s)+b.g(s)}{2}ds
\end{aligned}
$$

Essentially , this shows that if we are able to obtain an expression for the local and global bests, we can come up with an algebraic expression which will , in turn

, be a solution to the dynamics.

To find out the loci they follow, let's consider some special cases.

6.1 Expressions of global and local bests:

6.1.1 Solitary Particle:

In case of a solitary particle in the swarm, the dynamics changes drastically. This is the assumption we initially took to linearize the system. In this case the global best is always equal to the local best and current position of the particle , i.e.

$$x(t) = l(t) = g(t) \tag{6.2}$$

Thus,from equation 4.3, the state equation of the system with the solitary particle , becomes:

$$\begin{aligned} \dot{v_i(t)} &= -(1-\omega).v_i(t) + a.rand_1.(l_i(t) - x_i(t)) + b.rand_2.(g(t) - x_i(t)) \\ \Rightarrow \dot{v_g(t)} &= -(1-\omega).v_g(t);\ \text{since}\ , v_i(t) = v_g(t), x(t) = l(t) = g(t) \end{aligned}$$

Also,

$$\begin{aligned} v_i(t) &= \dot{x}_i(t) \\ \Rightarrow v_g(t) &= \dot{g}(t) \end{aligned}$$

From the two equations above, we can obtain the dynamics of the solitary particle, which is as follows:

$$\dot{y_g}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -(1-\omega) \end{bmatrix} y_g(t) \tag{6.3}$$

where $y_g(t) = \begin{bmatrix} g(t) \\ v_g(t) \end{bmatrix}$. Note that we did not consider the expected values of the variables here as we already have the time domain governing equations of PSO system from previous analyses. The solution of the dynamics is:

$$g(t) = g(0) + \frac{v_g(0)}{1-\omega}(1 - e^{-(1-\omega)t}) \tag{6.4}$$

and

$$v_g(t) = v_g(0).e^{-(1-\omega)t} \tag{6.5}$$

So, we can see , that the global best particle may change following exponential variation , in the form of $g(t) = p_1 + p_2.e^{-qt}$.

6.1.2 Exponential Variation of r(t):

Since we've seen $g(t)$ to follow an exponential variation , we assume $r(t)$ i.e $\frac{a.L(t)+b.g(t)}{2}$ to follow a similar variation. Thus,we assume $r(t) = \frac{a.L(t)+b.g(t)}{2} = p_1 + p_2 e^{-qt}$. Thus the equations obtained previously for $\mu_x(t)$ & $\mu_v(t)$, become:

$$\begin{aligned}
\mu_x(t) &= \frac{1}{\lambda_1 - \lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0) + \mu_v(0)) + e^{\lambda_2 t}(-\lambda_1\mu_x(0) + \mu_v(0))\Big\} \\
&+ \frac{1}{\lambda_1 - \lambda_2}\int_{t_0}^{t}\Big[e^{\lambda_1(t-s)} - e^{\lambda_2(t-s)}\Big]\frac{a.L(s) + b.g(s)}{2}ds \\
&= \frac{1}{\lambda_1 - \lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0) + \mu_v(0)) + e^{\lambda_2 t}(-\lambda_1\mu_x(0) + \mu_v(0))\Big\} \\
&+ \frac{1}{\lambda_1 - \lambda_2}\int_{t_0}^{t}\Big[e^{\lambda_1(t-s)} - e^{\lambda_2(t-s)}\Big](p_1 + p_2 e^{qs})ds \\
&= \frac{1}{\lambda_1 - \lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0) + \mu_v(0)) + e^{\lambda_2 t}(-\lambda_1\mu_x(0) + \mu_v(0))\Big\} \\
&+ \frac{p_1}{\lambda_2 - \lambda_1}\Big\{\frac{\lambda_2 - \lambda_1}{\lambda_1\lambda_2} - (\frac{e^{\lambda_1 t}}{\lambda_1} - \frac{e^{\lambda_2 t}}{\lambda_2})\Big\} \\
&+ \frac{p_2}{\lambda_2 - \lambda_1}\Big\{\frac{e^{\lambda_2 t} - e^{-qt}}{\lambda_2 + q} - \frac{e^{\lambda_1 t} - e^{-qt}}{\lambda_1 + q}\Big\}
\end{aligned}$$

In a similar fashion, we can show that,

$$\begin{aligned}
\mu_v(t) &= \frac{\lambda_2(-\lambda_2\mu_x(0) + \mu_v(0))}{\lambda_1 - \lambda_2}e^{\lambda_2 t} + \frac{\lambda_1(-\lambda_2\mu_x(0) - \mu_v(0))}{\lambda_1 - \lambda_2}e^{\lambda_1 t} \\
&+ \frac{p_1}{\lambda_2 - \lambda_1}\Big\{\frac{\lambda_2 - \lambda_1}{\lambda_1\lambda_2} - (\frac{e^{\lambda_1 t}}{\lambda_1} - \frac{e^{\lambda_2 t}}{\lambda_2})\Big\} \\
&+ \frac{p_2}{\lambda_2 - \lambda_1}\Big\{\frac{e^{\lambda_2 t} - e^{-qt}}{\lambda_2 + q} - \frac{e^{\lambda_1 t} - e^{-qt}}{\lambda_1 + q}\Big\}
\end{aligned}$$

6.1.3 No variation of $r(t)$ i.e. constant value:

Constant $r(t)$ is a special case of the exponential variation we previously considered i.e. $q = 0$. Thus the expressions obtained previously become:

$$\begin{aligned}\mu_x(t) = &\frac{1}{\lambda_1 - \lambda_2}\Big\{e^{\lambda_1 t}.(-\lambda_2\mu_x(0) + \mu_v(0)) + e^{\lambda_2 t}(-\lambda_1\mu_x(0) + \mu_v(0))\Big\}\\ &+ \frac{p_1}{\lambda_2 - \lambda_1}\Big\{\frac{\lambda_2 - \lambda_1}{\lambda_1\lambda_2} - (\frac{e^{\lambda_1 t}}{\lambda_1} - \frac{e^{\lambda_2 t}}{\lambda_2})\Big\}\\ &+ \frac{p_2}{\lambda_2 - \lambda_1}\Big\{\frac{e^{\lambda_2 t} - 1}{\lambda_2} - \frac{e^{\lambda_1 t} - 1}{\lambda_1}\Big\}\end{aligned}$$

and,

$$\begin{aligned}\mu_v(t) = &\frac{\lambda_2(-\lambda_2\mu_x(0) + \mu_v(0))}{\lambda_1 - \lambda_2}e^{\lambda_2 t} + \frac{\lambda_1(-\lambda_2\mu_x(0) - \mu_v(0))}{\lambda_1 - \lambda_2}e^{\lambda_1 t}\\ &+ \frac{p_1}{\lambda_2 - \lambda_1}\Big\{\frac{\lambda_2 - \lambda_1}{\lambda_1\lambda_2} - (\frac{e^{\lambda_1 t}}{\lambda_1} - \frac{e^{\lambda_2 t}}{\lambda_2})\Big\}\\ &+ \frac{p_2}{\lambda_2 - \lambda_1}\Big\{\frac{e^{\lambda_2 t} - 1}{\lambda_2} - \frac{e^{\lambda_1 t} - 1}{\lambda_1}\Big\}\end{aligned}$$

6.1.4 Expression for global best particle:

For global best particle,

$$x(t) = l(t) = g(t)$$

Thus, the equations become as follows:

$$\begin{aligned}\dot{v_g}(t) + (1 - \omega)v_g(t) = 0\\ \dot{g}(t) = v_g(t)\end{aligned}$$

Thus the state equations of the above system can be:

$$\dot{y_g}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -(1-\omega) \end{bmatrix} y_g(t)$$

where, $y_g(t) = \begin{bmatrix} g(t) \\ v_g(t) \end{bmatrix}$

Thus, the assumption of constant $r(t)$ is valid as in this case, we can see, for global best particle, $r(t) = 0$, irrespective of the objective function.

The solutions of the system defined by the above equation are as follows:

$$\begin{aligned} g(t) &= g(0) + \frac{v_g(0)}{1-\omega}\left(1 - e^{-t(1-\omega)}\right) \\ v_g(t) &= v_g(0)e^{-t(1-\omega)} \end{aligned}$$

6.1.5 Expression for local best particle:

For local best particle, we have,

$$x_i(t) = l_i(t)$$

Thus, the equations governing the dynamics are given by, as $x_i(t) = l_i(t)$:

$$\begin{aligned} \dot{v_i}(t) &= -(1-\omega).v_i(t) + a.rand_1.(l_i(t) - x_i(t)) + b.rand_2.(g(t) - x_i(t)) \\ \Rightarrow \dot{v_{l_i}}(t) &+ (1-\omega)v_{l_i}(t) + b.rand_2.l_i(t) = b.rand_2.g(t) \\ \dot{l_i}(t) &= v_{l_i}(t) \end{aligned}$$

Using expectation operator on both side of the equations, we obtain,

$$\begin{aligned} \dot{\mu_{v_L}}(t) + (1-\omega)\mu_{v_L}(t) + \frac{b}{2}L(t) &= \frac{b}{2}g(t) \\ \dot{L}(t) &= \mu_{v_L}(t) \end{aligned}$$

These equations, give us a dynamics as follows, which determines the locus of $L(t)$:

$$\dot{y_L}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{b}{2} & -(1-\omega) \end{bmatrix} y_L(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{b}{2}g(t)$$

where, $y_L(t) = \begin{bmatrix} L(t) \\ v_L(t) \end{bmatrix}$ The solution of this dynamics is straight forward and can be obtained in the similar fashion which we applied in case of original PSO dynamics. We can find, in the same way:

$$\begin{aligned} L(t) &= \frac{v_L(0) - \gamma_1 L(0)}{\gamma_2 - \gamma_1} e^{\gamma_2 t} + \frac{-v_L(0) + \gamma_2 L(0)}{\gamma_2 - \gamma_1} e^{\gamma_1 t} + \\ &\quad \frac{1}{\gamma_2 - \gamma_1} \int_0^t \Big(e^{\gamma_2 (t-s)} - e^{\gamma_1 (t-s)} \Big) r'(s) ds \\ v_L(t) &= \gamma_2 \frac{v_L(0) - \gamma_1 L(0)}{\gamma_2 - \gamma_1} e^{\gamma_2 t} + \gamma_1 \frac{-v_L(0) + \gamma_2 L(0)}{\gamma_2 - \gamma_1} e^{\gamma_1 t} \\ &\quad + \frac{1}{\gamma_2 - \gamma_1} \int_0^t \Big(\gamma_2 e^{\gamma_2 (t-s)} - \gamma_1 e^{\gamma_1 (t-s)} \Big) r'(s) ds \end{aligned}$$

where

$$\begin{aligned} \gamma_1, \gamma_2 &= \frac{-(1-\omega) \pm \sqrt{(1-\omega)^2 - 2b}}{2} \\ r'(t) &= \frac{b}{2} g(t) \end{aligned}$$

Since $g(t)$ follows an exponential variation i.e.

$$g(t) = g(0) + \frac{v_g(0)}{1-\omega} \Big(1 - e^{-t(1-\omega)} \Big)$$

putting value of $g(t)$ in $r'(t)$ and $L(t)$, we obtain:

$$r'(t) = \frac{b}{2} g(t) = \frac{b}{2} \Big\{ g(0) + \frac{v_g(0)}{1-\omega} \Big(1 - e^{-t(1-\omega)} \Big) \Big\}$$

and

$$\begin{aligned} L(t) &= \frac{v_L(0) - \gamma_1 L(0)}{\gamma_2 - \gamma_1} e^{\gamma_2 t} + \frac{-v_L(0) + \gamma_2 L(0)}{\gamma_2 - \gamma_1} e^{\gamma_1 t} \\ &\quad + \frac{1}{\gamma_2 - \gamma_1} \int_0^t \Big(e^{\gamma_2 (t-s)} - e^{\gamma_1 (t-s)} \Big) \frac{b}{2} \Big\{ g(0) + \frac{v_g(0)}{1-\omega} \Big(1 - e^{-s(1-\omega)} \Big) \Big\} ds \end{aligned}$$

Integrating we obtain,

$$
\begin{aligned}
L(t) = &\frac{1}{\gamma_2-\gamma_1}\left(\gamma_2 L(0) - v_L(0) - \frac{p_1}{\gamma_1} - \frac{p_2}{\gamma_2+q}\right)e^{\gamma_1 t} \\
&+ \frac{1}{\gamma_2-\gamma_1}\left(-\gamma_1 L(0) + v_L(0) + \frac{p_1}{\gamma_2} + \frac{p_2}{\gamma_2+q}\right)e^{\gamma_2 t} \\
&+ \frac{1}{\gamma_2-\gamma_1}\left(\frac{\gamma_2-\gamma_1}{\gamma_1\gamma_2}p_1 - \Big(\frac{1}{\gamma_2+q} - \frac{1}{\gamma_1+q}\Big)p_2 e^{-qt}\right)
\end{aligned}
$$

where,

$$
\begin{aligned}
q &= 1-\omega \\
p_1 &= \frac{b}{2}\left(g(0) + \frac{v_g(0)}{1-\omega}\right) \\
p_2 &= -\frac{b}{2}\frac{v_g(0)}{1-\omega}
\end{aligned}
$$

. Using $\gamma_1+\gamma_2 = -q$, we obtain,

$$
\begin{aligned}
L(t) = &\frac{1}{\gamma_2-\gamma_1}\left(\gamma_2 L(0) - v_L(0) - \frac{p_1}{\gamma_1} - \frac{p_2}{\gamma_2+q}\right)e^{\gamma_1 t} \\
&+ \frac{1}{\gamma_2-\gamma_1}\left(-\gamma_1 L(0) + v_L(0) + \frac{p_1}{\gamma_2} - \frac{p_2}{\gamma_1}\right)e^{\gamma_2 t} \\
&+ \frac{1}{\gamma_2\gamma_1}\left(p_1 + p_2 e^{-qt}\right)
\end{aligned}
$$

6.2 Combined Expression incorporating both L(t) and g(t):

Thus, $L(t)$ is seen to have the expression:

$$L(t) = m + m_1 e^{\gamma_1 t} + m_2 e^{\gamma_2 t} + m_3 e^{-qt} \tag{6.6}$$

where,

$$\begin{aligned}
m &= g(0) + \frac{v_g(0)}{1-\omega} \\
m_1 &= \frac{1}{\gamma_2 - \gamma_1}\Bigg(\gamma_2 L(0) - v_L(0) - \frac{p_1}{\gamma_1} + \frac{p_2}{\gamma_2}\Bigg) \\
m_2 &= \frac{1}{\gamma_2 - \gamma_1}\Bigg(- \gamma_1 L(0) + v_L(0) + \frac{p_1}{\gamma_2} - \frac{p_2}{\gamma_1}\Bigg) \\
m_3 &= -\frac{v_g(0)}{1-\omega}
\end{aligned}$$

So, combining the expressions of the terms $L(t)$ & $g(t)$ and putting them in the expression of $r(t)$, we get:

$$r(t) = \frac{a}{2}L(t) + \frac{b}{2}g(t)$$

$$\Rightarrow r(t) = \frac{a}{2}\Bigg(m + m_1 e^{\gamma_1 t} + m_2 e^{\gamma_2 t} + m_3 e^{-qt}\Bigg) + \frac{b}{2}\Bigg(g(0) + \frac{v_g(0)}{1-\omega}\Big(1 - e^{-t(1-\omega)}\Big)\Bigg)$$

Putting expression of $r(t)$ in $\mu_x(t)$ and $\mu_v(t)$, we obtain:

$$\begin{aligned}
\mu_x(t) = \frac{1}{\lambda_2 - \lambda_1}\Bigg(&\mu_v(0) - \lambda_1\mu_x(0) + \frac{m+p_1}{\lambda_2} + \frac{m_1}{\lambda_2 - \gamma_1} \\
&+ \frac{m_2}{\lambda_2 - \gamma_2} + \frac{m_3 + p_2}{\lambda_2 + q}\Bigg)e^{\lambda_2 t} \\
+ \frac{1}{\lambda_2 - \lambda_1}\Bigg(& - \mu_v(0) + \lambda_2\mu_x(0) - \frac{m+p_1}{\lambda_1} + \frac{m_1}{\lambda_1 - \gamma_1} \\
&+ \frac{m_2}{\lambda_1 - \gamma_2} + \frac{m_3 + p_2}{\lambda_1 + q}\Bigg)e^{\lambda_1 t} \\
+ \frac{m_1}{\lambda_2 - \lambda_1}&\Bigg(\frac{1}{\lambda_1 - \gamma_1} - \frac{1}{\lambda_2 - \gamma_1}\Bigg)e^{\gamma_1 t} \\
+ \frac{m_2}{\lambda_2 - \lambda_1}&\Bigg(\frac{1}{\lambda_1 - \gamma_2} - \frac{1}{\lambda_2 - \gamma_2}\Bigg)e^{\gamma_2 t} \\
+ \frac{m_3 + p_2}{\lambda_2 - \lambda_1}&\Bigg(\frac{1}{\lambda_1 + q} - \frac{1}{\lambda_2 + q}\Bigg)e^{-qt} + \frac{m + p_1}{\lambda_1\lambda_2}
\end{aligned}$$

and,

$$
\begin{aligned}
\mu_v(t) = {} & \frac{\lambda_2}{\lambda_2-\lambda_1}\left(\mu_v(0)-\lambda_1\mu_x(0)+\frac{m+p_1}{\lambda_2}+\frac{m_1}{\lambda_2-\gamma_1}+\frac{m_2}{\lambda_2-\gamma_2}+\frac{m_3+p_2}{\lambda_2+q}\right)e^{\lambda_2 t} \\
& +\frac{\lambda_1}{\lambda_2-\lambda_1}\left(-\mu_v(0)+\lambda_2\mu_x(0)-\frac{m+p_1}{\lambda_1}+\frac{m_1}{\lambda_1-\gamma_1}+\frac{m_2}{\lambda_1-\gamma_2}+\frac{m_3+p_2}{\lambda_1+q}\right)e^{\lambda_1 t} \\
& +\frac{m_1}{\lambda_2-\lambda_1}\left(\frac{1}{\lambda_1-\gamma_1}-\frac{1}{\lambda_2-\gamma_1}\right)e^{\gamma_1 t} \\
& +\frac{m_2}{\lambda_2-\lambda_1}\left(\frac{\lambda_1}{\lambda_1-\gamma_2}-\frac{\lambda_2}{\lambda_2-\gamma_2}\right)e^{\gamma_2 t} \\
& +\frac{m_3+p_2}{\lambda_2-\lambda_1}\left(\frac{\lambda_1}{\lambda_1+q}-\frac{\lambda_2}{\lambda_2+q}\right)e^{-qt}
\end{aligned}
$$

Thus we can write the PSO dynamics as follows:

$$
\begin{aligned}
\mu_x(t) &= A + A_1e^{\lambda_1 t}+A_2e^{\lambda_2 t}+A_3e^{\gamma_1 t}+A_4e^{\gamma_2 t}+A_5e^{-qt} \\
\mu_v(t) &= A_1e^{\lambda_1 t}+A_2e^{\lambda_2 t}+A_3'e^{\gamma_1 t}+A_4'e^{\gamma_2 t}+A_5'e^{-qt}
\end{aligned}
$$

where,

$$
\begin{aligned}
A &= \frac{m+p_1}{\lambda_1\lambda_2} \\
A_1 &= \frac{1}{\lambda_2-\lambda_1}\left(-\mu_v(0)+\lambda_2\mu_x(0)-\frac{m+p_1}{\lambda_1}+\frac{m_1}{\lambda_1-\gamma_1}+\frac{m_2}{\lambda_1-\gamma_2}+\frac{m_3+p_2}{\lambda_1+q}\right) \\
A_2 &= \frac{1}{\lambda_2-\lambda_1}\left(\mu_v(0)-\lambda_1\mu_x(0)+\frac{m+p_1}{\lambda_2}+\frac{m_1}{\lambda_2-\gamma_1}+\frac{m_2}{\lambda_2-\gamma_2}+\frac{m_3+p_2}{\lambda_2+q}\right) \\
A_3 &= \frac{m_1}{\lambda_2-\lambda_1}\left(\frac{1}{\lambda_1-\gamma_1}-\frac{1}{\lambda_2-\gamma_1}\right), A_3' = \frac{m_1}{\lambda_2-\lambda_1}\left(\frac{1}{\lambda_1-\gamma_1}-\frac{1}{\lambda_2-\gamma_1}\right) \\
A_5 &= \frac{m_3+p_2}{\lambda_2-\lambda_1}\left(\frac{1}{\lambda_1+q}-\frac{1}{\lambda_2+q}\right), A_5' = \frac{m_3+p_2}{\lambda_2-\lambda_1}\left(\frac{\lambda_1}{\lambda_1+q}-\frac{\lambda_2}{\lambda_2+q}\right) \\
A_4 &= \frac{m_2}{\lambda_2-\lambda_1}\left(\frac{\lambda_1}{\lambda_1-\gamma_2}-\frac{\lambda_2}{\lambda_2-\gamma_2}\right), A_4' = \frac{m_2}{\lambda_2-\lambda_1}\left(\frac{\lambda_1}{\lambda_1-\gamma_2}-\frac{\lambda_2}{\lambda_2-\gamma_2}\right)
\end{aligned}
$$

Chapter 7

Lyapunov Stability

7.1 Lyapunov Stability

Lyapunov , in 1892, showed that certain functions can be used to determine stability of an equilibrium point. [20]

Let $V : D \to \Re$ be a continuously differentiable function defined in a domain $D \in \Re^n$ that contains the origin.

The derivative of $V(x)$, along the trajectories of $\dot{x} = f(x)$,denoted by $V(x)$, is given by:

$$\begin{aligned} \dot{V}(x) &= \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \dot{x}_i = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(x) \\ &= \begin{bmatrix} \frac{\partial V}{\partial x_1} & \frac{\partial V}{\partial x_2} & \dots & \frac{\partial V}{\partial x_n} \end{bmatrix} \cdot \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \\ &= \frac{\partial V}{\partial x} f(x) \end{aligned}$$

The derivative of V, along the trajectories of the dynamical system , is thus, dependent of the system itself. So, $\dot{V}(x)$ will be different for different systems.

7.2 Lyapunov's Theorem

Let $x = 0$ be an equilibrium point for $\dot{x} = f(x)$, and $D \in \Re$ be a domain containing $x = 0$. Let $V : D \to \Re$ be a continuously differentible function such that

- $V(0) = 0$ and $V(x) > 0$ in $D - \{0\}$
- $\dot{V}(x) \leq 0$ in D

Then $x = 0$ is stable. Moreover, if $\dot{V}(x) < 0$ in $D - \{0\}$, $x = 0$ is asymptotically stable.

7.3 Lyapunov Function

Similarly, a continuously differentiable function $V(x)$ satisfying the conditions

- $V(0) = 0$ and $V(x) > 0$ in $D - \{0\}$
- $\dot{V}(x) \leq 0$ in D

is called Lyapunov function.

The surface $V(x) = c$ for some $c > 0$, is called a Lyapunov function or a level surface. The curves are shown in figure 7.3. The three ellipses represent three Lyapunov surfaces for three Lyapunov functions $V(x) = c_1$ or $V(x) = c_2$ or $V(x) = c_3$ respectively, while $c_1 < c_2 < c_3$.

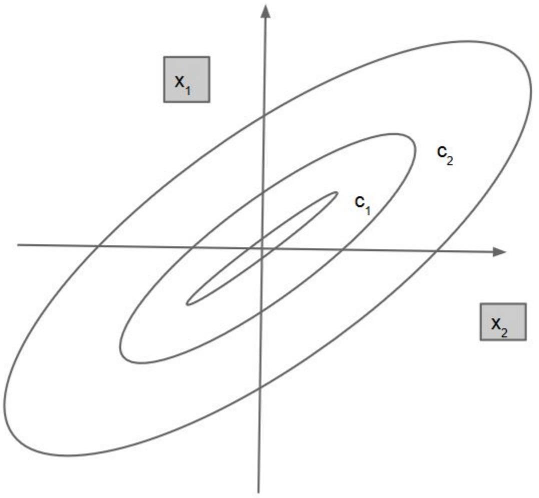


Figure 7.1: Level surfaces of Lyapunov Function

Now, the condition $\dot{V}(x) \leq 0$ implies that, whenever a trajectory of the function $f(x)$ crosses a Lyapunov function from the outside, in will never come out again.

When $V(x) < 0$, the trajectory moves from one Lyapunov surface to an inner Lyapunov surface with a smaller c. As c decreases , the Lyapunov surface $V(x) = c$ shrinks to the origin, showing that the trajectory approaches the origin as time passes by. But in case of $\dot{(V)}(x) \leq 0$,we can't be sure the trajectory will approach the origin.

7.4 Definiteness

7.4.1 Positive Definite & Semidefinite Functions

A function satisfying $V(0) = 0$, $V(x) > 0$ in $D - \{0\}$ is said to be positive definite. If it only satisfies a weaker condition $V(x) \geq 0$ in $D - \{0\}$, it's said to be positive semidefinite.

7.4.2 Negative Definite & Semidefinite Function

A function $V(x)$ is negative definite(or semidefinite) if $-V(x)$ is positive definite(semidefinite).

7.4.3 Indefinite

If $V(x)$ doesn't belong to any of the four categories mentioned above, it's indefinite.

Keeping these definitions in mind, we can state Lyapunov's theorem in a different manner.

7.5 Rephrased Lyapunov Theorem

The origin is stable if there is a continuous differentiable positive definite function $V(x)$ so that $\dot{V}(x)$ is negative semidefinite, and it's asymptotically stable if $\dot{V}(x)$ is negative definite.

The conditions stated in Lyapunov Theorem are only sufficient condition, which means if these conditions are satisfied , we can say the system is stable, corresponding to the particular Laypunov function candidate. The conditions are not at all necessary, which means , if they are not satisfied, nothing can be said about the stability of the system corresponding to other Lyapunov function candidates.

7.6 Generation of Lyapunov Functions

To apply the Lyapunov Theorem to determine the stability of an autonomous system , one needs to have a Laypunov function. In case of linear systems , we have well defined methods for generating such functions [22], whereas standard procedures for generating a Lyapunov function for non linear systems are being discovered recently.

7.7 Standard Method of Generating Lyapunov functions for linear systems

Let's consider the following autonomous linear system for the time being.

$$\dot{x} = Ax \tag{7.1}$$

This is a special case of equation 1.3, where A is not a function of x, and all elements of A are constants.

A very common scalar choice of $V(x)$, for which the sign definiteness can be checked easily, is the quadratic one, of the form:

$$V(x) = x'Px = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}x_i x_j \tag{7.2}$$

where P is a real symmetric matrix i.e. $p_{ij} = p_{ji}$.

From equation 7.2, the total derivative of $V(x)$ is:

$$\frac{dV}{dt} = \dot{x}'Px + x'P\dot{x}$$

As $\dot{x}' = (Ax)' = x'A'$,

$$\begin{aligned}\frac{dV}{dt} &= \dot{x}'Px + x'P\dot{x} \\ &= x'A'Px + x'PAx \\ &= x'(A'P + PA)x \\ &= -x'Qx\end{aligned}$$

where , $A'P + PA = -Q$ Thus,

$$V'(x) = -x'Qx \tag{7.3}$$

Since we have knowledge of A, we assume a Q which is suitable and try to find out about P, and thereby conclude the system stable if P is positive definite.

7.8 Settling Time

Given an initial condition, the time taken to reach the optimal solution , or the sufficient vicinity of it, is called the settling time.

In case of an LTI system, determination of settling time using Lyapunov function is trivial[22].

Let a parameter T ,be defined as follows

$$T_x = \frac{V(x)}{-\dot{V}(x)} \tag{7.4}$$

where

$$T = max(T_x)$$

Now,

$$
\begin{aligned}
&T_x \leq T \\
\Rightarrow &\frac{V(x(t))}{-\dot{V}(x(t))} \leq T \\
\Rightarrow &\frac{-\dot{V}(x(t))}{V(x(t))} \geq \frac{1}{T} \\
\Rightarrow &\int_{t=0}^{t=t_s} \frac{-\dot{V}(x(t))}{V(x(t))} dt \geq \int_{t=0}^{t=t_s} \frac{1}{T} dt \\
\Rightarrow &-\log \frac{V(x(t_s))}{V(x(0))} \geq \frac{t_s}{T} \\
\Rightarrow &\log \frac{V(x(0))}{V(x(t_s))} \geq \frac{t_s}{T} \\
\Rightarrow &\frac{V(x(0))}{V(x(t_s))} \geq e^{\frac{t_s}{T}} \\
\Rightarrow &V(x(t_s)) \leq V(x(0)) e^{-\frac{-t_s}{T}}
\end{aligned}
$$

So, this relation gives an upper bound on the value of $V(x(t_s))$ with $V(x(t))$ having an initial value, $V(x(0))$. Thus, if we specify the time t_s , we'll be able to know the upper bound at that time. So, if we specify the upper bound, we can calculate the time required for $V(x(t))$ to reach at or below that. This time is called the settling time. It is used to the system to reach at a certain level of vicinity of the chosen equilibrium point.

Since $V(x(t))$ can be chosen according to the wish of the user, as long as it is positive definite, and so can be $\dot{V}(x(t))$, as long as it can be negative definite. If it is negative semidefinite , the equilibrium point is not asymptotically stable, thus the question of settling time, becomes meaningless.

Suppose,

$$
\begin{aligned}
V(x(t)) &= x'Px \\
\dot{V}(x(t)) &= -x'Qx
\end{aligned}
$$

Where Q and P are both positive definite matrices. $\therefore$

$$\begin{aligned} T &= max(\frac{V(x(t))}{-\dot{V}(x(t))}) \\ &= max(\frac{x'Px}{x'Qx}) \end{aligned}$$

As the relative shape and size of $V(x(t))$ and $V'(x(t))$ remains same throughout the space, we can set $x'Qx = 1$. Thus, $T = max(x'Px)$. This is the shape of an optimization problem as follows:

$$\begin{aligned} &max(x'Px) \\ &x'Qx = 1 \end{aligned}$$

Using the technique of Lagrangian multiplier , the problem becomes

$$max(x'Px - \lambda x'Qx)$$

where λ is the Lagrangian multiplier. Equating the derivative of the function $(x'Px - \lambda x'Qx)$, we get,

$$(P - \lambda Q)x = 0 \tag{7.5}$$

Premultiplying by x', we get

$$\begin{aligned} &x'Px = x'\lambda Qx \\ \Rightarrow\ &x'Px = \lambda x'Qx \\ \Rightarrow\ &x'Px = \lambda \end{aligned}$$

since λ is a scalar and $x'Qx = 1$.

Now, we can see from equation 7.5 that λ is an eigenvalue of the matrix $Q^{-1}P$. And also , $x'Px$ is maximum , if λ is maximum. Thus , $T = max(x'Px) =$ maximum eigenvalue of $Q^{-1}P$[22].

7.9 Calculation of Settling time of Linearized PSO dynamics

In this case , $A = \begin{bmatrix} 0 & 1 \\ -\frac{a+b}{2} & -(1-\omega) \end{bmatrix}$. Let's assume a positive definite Q arbitrarily as $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Solving the equation $A'P + PA = -Q$, where $a = 0.15, b = 0.25, \omega = -0.05$ we obtain,

$$P = \begin{bmatrix} 3.1964 & 2.500 \\ 2.5000 & 2.8571 \end{bmatrix}$$

. Now, the function $V(x)$ becomes

$$\begin{aligned} V(x) &= x'Px \\ &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 3.1964 & 2.500 \\ 2.5000 & 2.8571 \end{bmatrix} \begin{bmatrix} x-1 \\ x_2 \end{bmatrix} \\ &= 3.1964x_1^2 + 5x_1x_2 + 2.8571x_2^2 \end{aligned}$$

. From the analysis of the previous section , T = maximum eigenvalue of the matrix $Q^{-1}P. \therefore T = max(0.5210, 5.5325) = 5.5325$. Now , assuming the initial conditions to be $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.4248 \\ -0.3778 \end{bmatrix}$, the values are arbitrarily taken the values of one of the first iterations during the simulations, and we want the settling time for the variables to go nearby $\begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix}$, $V(\begin{bmatrix} 0.4248 \\ -0.3778 \end{bmatrix}) = 0.1822$, whereas

$V(\begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix}) = 0.0276$. Now, the settling time is given as

$$\begin{aligned} t_s &\leq -\lambda \log(\frac{V(x(t_s))}{V(x(0))}) \\ &= -5.5325 \log(\frac{0.0276}{0.1822}) \\ &= 10.4414 \end{aligned}$$

. So, at most 10.4414 seconds are required to get at the point $\begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix}$.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This current work has studied the dynamics of PSO which can be extended to any other form of PSO. It shows that the nature of the dynamics depends on the parameter settings, which also determines the stability of the system. A general control theoretic approach has been taken to conclude those results. It has also been shown that linearization is possible in case of the non linear system dynamics. And also, just like any other non linear dynamics, it follows the linearized dynamics in the vicinity of the equilibrium points around which the system was linearized. The lyapunov stability criterion is applied in case of the linear dynamics and settling time was calculated. The settling time, thus gives us a measure of the speed of convergence of the algorithm, which in turn , enables us to compare it with other algorithms.

Also, in practical applications of PSO it is sometimes desired that the algorithm must have some exploration power before it starts to converge. The exploration power can be achieved by having an unstable dynamics and by modulating the values of the control parameters throughout the iteration procedure, one can achieve desired amount of exploration at a desired time instance.

A general solution of the non linear dynamics is obtained, so that the effect of both social attraction co-efficient and self attraction co-efficient can be studied and compared.

8.2 Future Work

Hafstein [15] recently obtained a procedure for obtaining a piecewise affine Lyapunov function, for determining the stability of the non linear systems. His procedure iterates over a set of possible Lyapunov function candidates and chooses the best one. Although this procedure doesn't give us an algebraic expression for the function, it does give us the value of the function at certain points. Applying Hafstein's technique for obtaining a suitable Lyapunov function can be one obvious extension of this work.

The extension of similar treatments to other evolutionary algorithms, like Differential Evolution (DE) can be another step forward. This approach to Evolutionary Algorithms actually help us find the strong mathematical background they have underneath and thus solve real life problems, more accurately and efficiently.

Bibliography

[1] Metaheuristics image. https://en.wikipedia.org/wiki/Metaheuristic#/media/File:Metaheuristics_classification.svg.

[2] David K Arrowsmith and Colin M Place. *An introduction to dynamical systems*. Cambridge University Press, 1990.

[3] Tim M Blackwell. Particle swarms and population diversity. *Soft Computing*, 9(11):793–802, 2005.

[4] Bernhard Brandstätter and Ulrike Baumgartner. Particle swarm optimization-mass-spring system analogon. *Magnetics, IEEE Transactions on*, 38(2):997–1000, 2002.

[5] Daniel Bratton and James Kennedy. Defining a standard for particle swarm optimization. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 120–127. IEEE, 2007.

[6] Emilio F Campana, Giovanni Fasano, Daniele Peri, and Antonio Pinto. Particle swarm optimization: Efficient globally convergent modifications. In *Proceedings of the III European conference on computational mechanics, solids, structures and coupled problems in engineering, Lisbon, Portugal*, 2006.

[7] Emilio F Campana, Giovanni Fasano, and Antonio Pinto. Dynamic system analysis and initial particles position in particle swarm optimization. In *IEEE Swarm Intelligence Symposium*, 2006.

[8] Xin Chen and Yangmin Li. A modified pso structure resulting in high exploration ability with convergence guaranteed. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(5):1271–1289, 2007.

[9] Ying-ping Chen and Pei Jiang. Analysis of particle interaction in particle swarm optimization. *Theoretical Computer Science*, 411(21):2101–2115, 2010.

[10] Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1):58–73, 2002.

[11] Hassan M Emara, Abdel Fattah, et al. Continuous swarm optimization technique with stability analysis. In *American Control Conference, 2004. Proceedings of the 2004*, volume 3, pages 2811–2817. IEEE, 2004.

[12] Juan Luis Fernandez-Martinez and Esperanza Garcia-Gonzalo. Stochastic stability analysis of the linear continuous and discrete pso models. *Evolutionary Computation, IEEE Transactions on*, 15(3):405–423, 2011.

[13] Hao Gao and Wenbo Xu. A new particle swarm algorithm and its globally convergent modifications. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(5):1334–1351, 2011.

[14] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[15] Sigurdur Freyr Hafstein. *An algorithm for constructing Lyapunov functions.* Department of Mathematics, Texas State University, 2007.

[16] Ming Jiang, YP Luo, and SY Yang. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, 102(1):8–16, 2007.

[17] Visakan Kadirkamanathan, Kirusnapillai Selvarajah, and Peter J Fleming. Stability analysis of the particle dynamics in particle swarm optimizer. *Evolutionary Computation, IEEE Transactions on*, 10(3):245–255, 2006.

[18] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.

[19] James Kennedy and Rui Mendes. Population structure and particle swarm performance. 2002.

[20] Hassan Khalil. Nonlinear systems. 2002.

[21] Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984.

[22] Ajit K Mandal. *Introduction to Control Engineering: Modeling, Analysis and Design*. New Age International, 2006.

[23] JL Martínez and E García Gonzalo. The generalized pso: a new door to pso evolution. *Journal of Artificial Evolution and Applications*, 2008:5, 2008.

[24] Rui Mendes and James Kennedy. Avoiding the pitfalls of local optima: how topologies can save the day. 2003.

[25] Rui Mendes, James Kennedy, and José Neves. The fully informed particle swarm: simpler, maybe better. *Evolutionary Computation, IEEE Transactions on*, 8(3):204–210, 2004.

[26] Ender Ozcan and Chilukuri K Mohan. Analysis of a simple particle swarm optimization system. *Intelligent engineering systems through artificial neural networks*, 8:253–258, 1998.

[27] Ender Ozcan and Chilukuri K Mohan. Particle swarm optimization: surfing the waves. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE, 1999.

[28] Riccardo Poli. Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *Evolutionary Computation, IEEE Transactions on*, 13(4):712–721, 2009.

[29] Nayan R Samal, Amit Konar, Swagatam Das, and Ajith Abraham. A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1769–1776. IEEE, 2007.

[30] Dirk Sudholt and Carsten Witt. Runtime analysis of a binary particle swarm optimizer. *Theoretical Computer Science*, 411(21):2084–2100, 2010.

[31] Ioan Cristian Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters*, 85(6):317–325, 2003.

[32] Frans Van Den Bergh. *An analysis of particle swarm optimizers.* PhD thesis, University of Pretoria, 2006.

[33] Gerhard Venter and Jaroslaw Sobieszczanski-Sobieski. Particle swarm optimization. *AIAA journal*, 41(8):1583–1589, 2003.