ISSN : 2165-4069(Online)
ISSN : 2165-4050(Print)

IJARAI

International Journal of
Advanced Research in Artificial Intelligence

Volume 4 Issue 5

www.ijarai.thesai.org

A Publication of
The Science and Information Organization

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ARTIFICIAL INTELLIGENCE

THE SCIENCE AND INFORMATION ORGANIZATION

www.thesai.org | info@thesai.org

Editorial Preface

From the Desk of Managing Editor…

Artificial Intelligence is hardly a new idea. Human likenesses, with the ability to act as human, dates back to Geek mythology with Pygmalion's ivory statue or the bronze robot of Hephaestus. However, with innovations in the technological world, AI is undergoing a renaissance that is giving way to new channels of creativity.

The study and pursuit of creating artificial intelligence is more than designing a system that can beat grand masters at chess or win endless rounds of Jeopardy!. Instead, the journey of discovery has more real-life applications than could be expected. While it may seem like it is out of a science fiction novel, work in the field of AI can be used to perfect face recognition software or be used to design a fully functioning neural network.

At the International Journal of Advanced Research in Artificial Intelligence, we strive to disseminate proposals for new ways of looking at problems related to AI. This includes being able to provide demonstrations of effectiveness in this field. We also look for papers that have real-life applications complete with descriptions of scenarios, solutions, and in-depth evaluations of the techniques being utilized.

Our mission is to be one of the most respected publications in the field and engage in the ubiquitous spread of knowledge with effectiveness to a wide audience. It is why all of articles are open access and available view at any time.

IJARAI strives to include articles of both research and innovative applications of AI from all over the world. It is our goal to bring together researchers, professors, and students to share ideas, problems, and solution relating to artificial intelligence and application with its convergence strategies. We would like to express our gratitude to all authors, whose research results have been published in our journal, as well as our referees for their in-depth evaluations.

We hope that this journal will inspire and educate. For those who may be enticed to submit papers, thank you for sharing your wisdom.

Editor-in-Chief
IJARAI
Volume 4 Issue 5 May 2015
ISSN: 2165-4069(Online)
ISSN: 2165-4050(Print)

Editorial Board

Reviewer Board Members

CONTENTS

The Influence of Stubborn Agents in a Multi-Agent Network for Inter-Team Cooperation/Negotiation

Eugene S. Kitamura
Department of Computer Science,
National Defense Academy
Yokosuka, Japan

Akira Namatame
Department of Computer Science,
National Defense Academy
Yokosuka, Japan

***Abstract*—When teams interact for cooperation or negotiation, there are unique dynamics that occur depending on the conditions. In this paper, a multi-agent system is used under the restrain of a network structure to model two teams of agents interacting for a common consensus, however with the presence of stubborn agents. The networks used were a minimum dumbbell network and two scale-free networks joined together. The network topology, which is a global characteristic, along with the presence of conflicting stubborn agents, can cause various conditions that affect teamwork in cooperation or negotiation. Notable characteristics revealed are boundary role persons (BRPs), lack of unity, need for a third party moderator, coalition formation, and loyalty of the BRP dependent on the distance from the core ideology of the team. Both local and global characteristics of network structures contribute to such phenomenon. The modeling method and corresponding simulation results provide valuable insight for predicting possible social dynamics and outcome when planning cooperation/negotiation tactics.**

Keywords—Multi-agent system; consensus problem; stubborn agents; complex network; dumbbell network; Laplacian matrix; boundary role person; coalition formation

I. INTRODUCTION

A. Team dynamics and consensus

The success of an organization depends on how effective its team of agents operate, whether in business or in other social settings. Thus, it is natural for organizational managers to have an interest in understanding group dynamics. A team, as opposed to an individual, is a conglomeration of diverse talents and perspectives, and allows distribution of tasks for efficient operations [1, 2]. Formation of a robust team with diverse agents is especially valued today in business to cope with rapidly changing technological and global markets [3, 4]. However, because of its collective nature of talents and personalities, a team of agents may be exposed to internal conflict and noise in addition to external ones during cooperation or negotiation with another team of agents. Such perturbations can result in unintended consensus dynamics within the team. How does the formation of consensus opinion depend on the locations of stubborn agents in the social network? We would like to be able to predict the outcomes such that we can control or avoid certain situations during the formation of consensus.

To have a holistic view of team dynamics, a manager must consider the *micro* and *macro* properties of a team. *Micro properties* considered important are *planning*, which is sub-dividing tasks among agents, *coordination*, which is the synchronization of agent actions and its continuous monitoring for assessment, and *communication*, which is information exchange among agents [5]. These qualities are micro properties since they are defined for and executed by the agents that form a team. On the other hand, *macro properties* are the network structure that embeds the agent interactions or patterns of relationships within a team [6, 7]. With such network topological aspect, the team dynamics is not only determined by agent characteristics or local interaction rules, but also determined by a global structural restraint. This network formed by the agent team can be considered a macro property since this property ignores the characteristics of each agent and it affects the entire community. For managers and team members to understand and predict team consensus dynamics, they need to observe and evaluate both their micro and macro properties. Failure of such considerations may result in disagreements and formations of coalitions that may impede the team's initial goal.

Consensus dynamics in general have been studied in computer science and control theory for a long time [8]. In a network of agents, consensus is formed when all of the agents agree to a certain parameter state. A consensus algorithm performs a consensus procedure where agents exchange their parameter status among its local neighbors in the agent network so that they may eventually reach an agreement. The analytical foundation of consensus problems for networked systems was presented by Olfati-Saber and colleagues [9]. The consensus problem is also related to synchronization. Synchronization phenomena are seen in diverse settings such as neurons firing, laser cascades, biological cycles, opinion formation, and in chaotic dissipative systems in general. Many such phenomena in nature are realized by synchronization in a constructive manner. However, forming synchronization in a harmful context can cause detrimental effects such as collapse of bridges or causing epileptic seizures [10, 11]. In order to manage such destructive outcomes by synchronization, Louzada [12] studied the use of contrarians to suppress undesired synchronization. Contrarians systematically dephase from the oscillation of their nearest neighbors. Louzada compared the use of contrarians with access to local and global information and concluded that contrarians with local interactions are enough for the most efficient influence. Additionally, when the interacting neighbor number (degree distribution) is relatively even and contrarians are placed at

highly connected nodes, the synchronization dephasing performance is significantly improved.

In this paper, instead of contrarians that desynchronize with its neighbors, the presence of stubborn agents that refuse to form consensus with the rest of the agents in the network is considered. Consensus formation of agents under the restriction of a network is observed. In particular, the network used is structured so that it can represent two teams interacting. Two stubborn agents with opposing opinions are used to represent a disagreement in the network. The terms *agent* and *node* are often used interchangeably, where an *agent* may have a stronger social/multi-agent system aspect whereas a *node* may have a complex network connotation.

B. Consensus formation with stubborn agents

Studies of consensus and synchronization under a network structure is important for understanding its dynamics as seen in the previous section. However, there is no guarantee that all of its agents in the network will cooperate. Some agents may lead the team to form a consensus by influencing the rest of the agents or some agents may mislead the consensus formation to an unintended final state. These situations may occur in diverse scenes such as socio-economic situations [13], rendezvous strategies [14], average consensus [15], and sensor deployment [16]. Gupta [17] studied possible scenarios of agent "failure" in the context of distributed algorithms used in the above research presented. The first failure condition is a stopping failure [18] where an agent blacks out and stops communicating with the other agents. The second failure condition is when an agent value becomes stuck at a fixed state. Fagnani [19] showed that if the rest of the agents are non-stubborn, then the agents would converge to this fixed value agent. The third failure condition is when an agent continuously changes its state to erroneous values at every time step, either intentionally or unintentionally [18, 20].

In a social context of stubborn agents, Acemoglu [21] studied the spread of *misinformation* by using "forceful" agents in an agent value averaging model. Forceful agents are not completely stubborn, but under particular conditions they have a strong influence on some of their neighbors such that the terminal consensus value is diverted from the original consensus value without the forceful agents. Spread of misinformation is quantified by measuring the magnitude of this divergence. Instead of forceful agents, Yildiz [22] presented a consensus behavior study with two stubborn agents with opposing opinions with different fixed agent values, using a classical voter model [23, 24]. In this model, the stubborn agent does not affect all of its neighbors, but rather chooses one neighbor randomly, and instead of taking an average value between the two agents, the neighbor agents adopt their neighbor's value. They found that with the presence of opposing stubborn agents, the opinions among the agent society disagree and fluctuate. Finally, Acemoglu [25] uses an inhomogeneous stochastic gossip model of communication. The agents update their belief as a convex combination of their own belief and the belief of their neighbor at the same time step. In addition to finding that consensus process fluctuates and never converges, they demonstrated that in a general network topology the intermediate agents between the stubborn agents take terminal values which are linear interpolations of the two stubborn agents' beliefs.

In the three above mentioned investigations, they consider dumbbell (barbell) graphs to observe the influence of either single, double, and triple stubborn agents on the rest of the non-stubborn agents. Their dumbbell graphs have two or three cliques of arbitrary agent number. Two small interacting teams can be modeled with a dumbbell graph with complete graphs connected by bridging agents and a link. The smallest complete graphs with three agents are a sufficient condition for a negotiation team [26], since working in small teams allow more flexibility, agility, and adaptability [27, 28]. Modeling of interactions between larger organizations may not be so simple due to its hierarchical structure or the involvements of many sub-divisions in the organization. However, considering that even large organizations would have a small scale working unit or a representative team for negotiations [29], a dumbbell teamwork interaction model is a reasonable and practical consideration. Due to the linking property of a dumbbell graph, special roles are played by bridging agents and link. In graph theoretical terms, these bridging agents have a higher betweenness centrality [30]. In a team cooperation or negotiation context, these bridging agents are called boundary role persons (BRPs) [31] and have a unique role in the team [32]. Later, a modeling of larger teams is also considered by using two scale-free networks connected.

With the presence of one or two stubborn agents, the following questions are investigated: what is the overall dynamics of the society of agents with the presence of stubborn agents? How opinion dynamics are affected with the presence of two stubborn agents holding completely opposite opinions? These are investigated under the constraint of a network structure. First, the time progression of the consensus formation process or stabilization process (if the dynamics is non-convergent) for a minimum dumbbell graph is demonstrated. The time progression shows the various patterns of approach resulting in diverse final outcomes depending on the location of the stubborn agents. Second, the notion of small team dynamics is extended to a team with a larger population and a greater distance of the BRP from the core ideology (stubborn agent) of its team. Two identical scale-free networks with a larger population are used instead of minimum complete graphs. The final states settle as a linear interpolation of agent locations as observed in [25]. However, the time progression shows that the consensus is formed among the local tree modules first, then a global steady state is reached. The presence of two stubborn agents results in the formation of coalitions. The model also supports that the farther away physically and psychologically the BRP is from the core team ideology (stubborn agent), the BRP agent opinion becomes closer to its opponent group [33].

II. ANALYTICAL FOUNDATION

A. Consensus protocol

The average consensus problem makes use of spectral graph theory and matrix theory [15, 34, 35]. Consider a symmetric (undirected) connected graph $G = (V, E)$ which is the network topology that restricts the agent interactions, where

V is a set of nodes and E is a set of edges. The linear continuous-time consensus protocol used for the network agent dynamics is [15, 35],

$$\dot{x}_i = \sum_{j \in N_i} \alpha_{ij} \left(x_j(t) - x_i(t) \right), \quad (1)$$

where x_i and x_j is the state value of agent i, and its network neighbor j respectively, α_{ij} is the weight of agent i on agent j, N_i is the set of neighbors connected to agent i, and t is time or iterations. Here, $\alpha_{ij} = \alpha_{ji}$ for all i, j, since the graph G is undirected. Consensus is achieved when the agent values converges to a common value based on (1), i.e.,

$$x_1 = x_2 = \cdots = x_n = \alpha, \quad (2)$$

where n is the total number of agents in the network. When there are no stubborn agents or other interferences in the agent network, the value taken by (2) can be expressed by the following equation,

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} x_i\,(0), \quad (3)$$

where $x_i(0)$ is the initial agent values at t=0. In other words, without any bias the consensus converges asymptotically to the average of the initial values. The Laplacian matrix L of the graph network G is explained below. The Laplacian matrix is defined as the following,

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \quad (4)$$

where $\mathbf{D} = \text{diag}(d_1, d_2, \cdots, d_n)$ is the diagonal matrix with elements $d_i = \sum_j a_{ij}$ and $\mathbf{A}$ is the binary adjacency matrix ($n \times n$ matrix) with elements a_{ij} for all i, j where a_{ij} is 1 if agent i and agent j is connected or 0 if they are disconnected. Then the dynamics of the system in (1) can be expressed as

$$\dot{x} = -\mathbf{L}x(t). \quad (5)$$

B. Stubborn agents

The presence of stubborn agents affects the formation of consensus, and the deviation of this consensus opinion dynamics depends on the network structure and the location of the stubborn agents on that structure. Fagnani [19] shows the analytical framework of consensus dynamics over networks including regular agents and stubborn agents. Consider a symmetric connected graph $G = (V, E)$ introduced earlier. Separate $V = S \cup R$ with the understanding that agents in S are stubborn agents not changing their state while agents in R are regular agents whose state modifies with time according to the consensus dynamics. With the presence of stubborn agents, the whole consensus dynamics can be described by the relation

$$x(t + 1) = P\,x(t). \quad (6)$$

When the elements in V are ordered in such a way that elements in R come first, the matrix P will exhibit the block structure:

$$P = \begin{bmatrix} Q_1 & Q_2 \\ 0 & I \end{bmatrix}. \quad (7)$$

By splitting the state vector accordingly, $x(t) = (x_R(t), x_S(t))$, the dynamics shown below is obtained.

$$x_R(t + 1) = Q_1 x_R(t) + Q_2 x_S(t)$$

$$x_S(t + 1) = x_S(t) \quad (8)$$

Notice that Q_1 is a sub-stochastic matrix, i.e., all row sums are less than or equal to 1. There is at least one row whose sum is strictly less than one, which is the row corresponding to a regular agent connected to a stubborn one. Using the connectivity of the graph, this easily implies that there exists t such that $(Q_1)^t$ has the property that all its rows have sum strictly less than one. This immediately yields that the matrix is asymptotically stable. Therefore,

$$x_R(t) \to x_R(\infty) \text{ for } t \to +\infty$$

with the limit opinions satisfying the relation which is equivalent to

$$x_R(\infty) = Q_1 x_R(\infty) + Q_2 x_S(0)$$

$$x_R(\infty) = (I - Q_1)^{-1} Q_2 x_S(0). \quad (9)$$

Regular agents settle asymptotically to opinions that are the convex combinations of the opinions of stubborn agents. The above analysis shows, that if all stubborn agents are in the same state x, for instance there is just one stubborn agent, then, consensus is reached by all agents in the opinion of the stubborn agent x. However, typically, consensus is not reached. Few examples modeled by simulation are discussed below. The initial state value of each agent is set as,

$$x_i(0) = i \ (i = 1, 2, \cdots, N). \quad (10)$$

When each agent updates her state value according to (1), the state of each agent converges to a constant value, which is the average of the initial values of all agents. The consensus dynamics in (1) converge to the average of the initial states of all agents without stubborn agents. Additionally, the consensus dynamics profile is different depending on the location of the two stubborn agents.

C. Networks

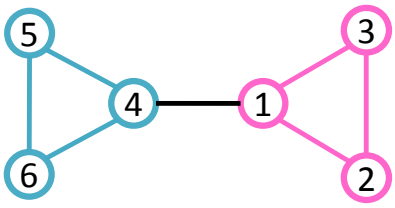


Fig. 1. A diagram of a dumbbell graph. Each clique is composed of 3 agents. The graph contains total of 6 nodes and 7 links. Since each clique consists of a minimum complete graph with 3 nodes, this network is considered a minimum dumbbell network

In the first experiment, we use a dumbbell network. A dumbbell network is a structure with two cliques connected by a single link (Fig. 1). "Simplest" here means the network consists of minimum number of nodes. The definition of a negotiating team is a group of two or more interdependent persons who collaborate to achieve a mutual goal through negotiation and they all attend the bargaining table [36]. The cliques used here consist of three nodes all connected to each other to form a complete graph. One arbitrarily node is chosen from each clique and these nodes are connected by a bridging link. These bridging nodes play a unique role in that they have an absolute influence on their own clique and the greatest influence to the opposite clique when stubborn agents are located at these bridging nodes.

In the second experiment, two scale-free networks were connected. This network was generated to model a greater number of nodes and greater distance between the stubborn agent and the bridge node. Two identical scale-free networks with 50 nodes were generated and the nodes with the least closeness centrality or the most "remote" nodes (Fig. 4, node 50 and 100) were connected. The node chosen for each stubborn agent is the highest alpha-centrality [37] for each scale-free network (nodes 1 and 51). The overall network forms a line network with modular tree structures attached to the nodes on the main line network. A line network is the simplest structure to model the "psychological distance" between the organizational core and the boundary. The entire network is symmetric since two identical networks were connected at identical locations.

III. RESULTS

First, the results from the simplest case of a dumbbell network are presented. On the dumbbell network, where agents are grouped with their clique network, local consensus is promoted in each clique network at first. After that, the global consensus is formed via the bridge link as seen in Fig. 2. Progressively increasing and evenly spaced initial values are assigned to the agents from agent 1through 6 as seen in Fig. 2, except for stubborn agents. Stubborn agents are assigned a node value of either 0 or 100, two extremely opposite values. Without stubborn agents, the agent converges asymptotically, eventually reaching a consensus value of the average of agent values as indicated in (3). Smaller values are assigned to agents in the violet color clique and larger values are assigned to agents in the blue color clique in Fig. 1.

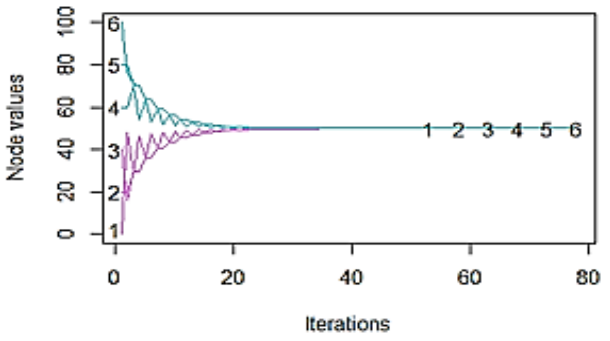


Fig. 2. A consensus progression diagram for the dumbbell graph in Fig. 1. The initial values assigned to agents in each clique are 0 to 50 to one clique and the 51 to100 to the other clique

With the presence of stubborn agents, three types of consensus behavior were noticed for a dumbbell network topology with single and double stubborn agents. The first type is for a single stubborn agent located at node 1 in Fig. 1 (case 1, Fig. 3(a)). This is a trivial solution where all of the agent opinions converge to the stubborn agent, regardless of the location of the stubborn agent. Such dynamics has been analytically modeled by Fagnani [19]. The second type is for two stubborn agents in the same clique. There are two possibilities for this type. The first possibility is when the stubborn agents are located at nodes 2 and 3 so none of the stubborn agents are the bridge agent (case 2, Fig. 3(b)). The rest of the regular agents from both cliques converge to an intermediate value. For the second possibility, the stubborn agents are located at nodes 1 and 2, one of the stubborn agents being a bridge agent (case 3, Fig. 3(c)). The regular agent (agent 3) in the same clique will converge to an intermediate value between the two stubborn agents and all of the agents in the opposite clique without stubborn agents will converge to the bridging stubborn agent node 1. In this case, the regular agents trapped between the stubborn agents converge to an average value between the two stubborn agent values.

The third type is again for two stubborn agents, but one located in each clique. In general, the regular agents in a clique will approach the stubborn agent in their respective cliques, however, under the influence of the stubborn agent of the opposite clique. When one of the stubborn agents is located at node 1 bridge agent (case 4, Fig. 3(d)), this stubborn agent shuts out the rest of the agents in its clique from any influence from the opposite clique. One could say that the stubborn agent at the bridge node "dominates" its clique. Similarly, this bridging stubborn agent will influence the agents in the opposite clique, where its influence will be the strongest with the bridging regular agent node 4, then with indirect and reduced influence with agents beyond the bridge agent of the opposite clique (agent 6), except for the stubborn agent (agent 5). In this case, the regular agents with indirect influence converge to a value between its stubborn agent (agent 5) and the bridging agent (agent 4) as in case 3.

When the two stubborn agents are located at the bridge nodes of their respective cliques, agents 1 and 4 (case 5, Fig. 3(e)), the regular agents will converge to the stubborn agent of their own clique. The regular agents in one clique will never reach a consensus with the regular agents in the opposite clique and vice versa because all influence from the opposite clique is blocked by the stubborn agent at the bridge node of its own clique.

If none of the bridge agents are stubborn (case 6, Fig. 3(f)), the regular agents are not topologically restricted to the influence of a particular stubborn agent. The bridge agents receive the largest influence of the opposite clique and the rest of the regular agents converge to a value between their bridge agent and the stubborn agent in their clique. The regular agents do not converge with the stubborn agent in their clique because of the influence of the stubborn agent in the opposite clique. In general, when the number of agents in the cliques increases while maintaining a complete graph structure, the values of regular agents within the clique tend to attract or "bond" closer together.

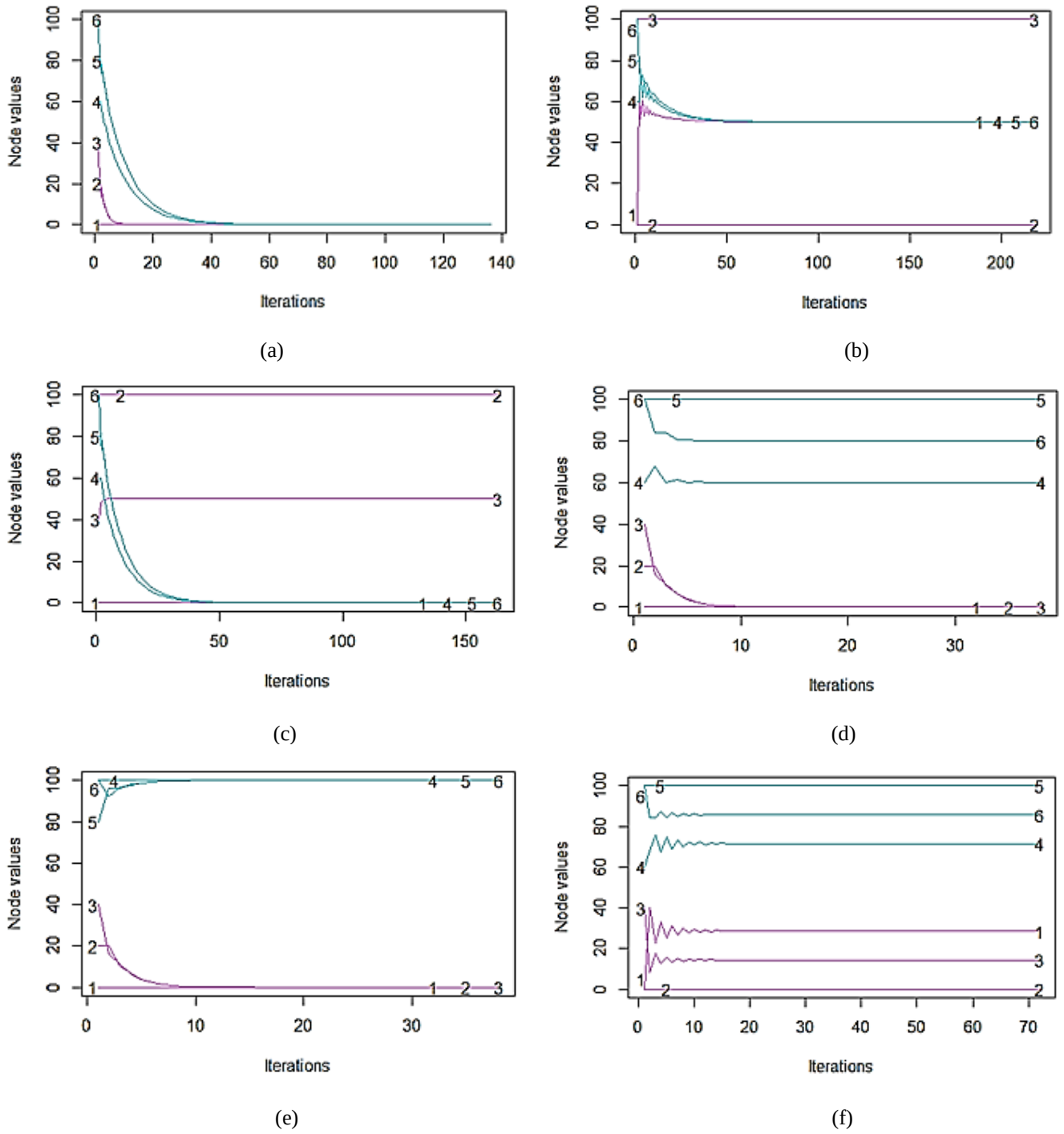


Fig. 3. Graph with the change in the agent value. *x*-axis is time step and *y*-axis is node value. Node number for stubborn agents are (a) node 1, (b) nodes 2 and 3, (c) nodes 1 and 2, (d) nodes 1 and 5, (e) nodes 1 and 4, and (f) nodes 2 and 5. The node numbers are written on the graph lines

Not only that, the value of the regular agents, except for the bridging agents, converge to a single value. The bridging agents of the opposite cliques attract each other so they do not converge completely with the rest of the regular agents in their respective clique. This is also seen in case 4 (Fig. 3(d)). The following is the result from the scale-free network structure (Fig. 4). The number of time steps it took for the agent values to stabilize ($\Delta x_i=10^{-9}$, which is the difference in the values between two time steps) is close to 6,300 time steps (Fig. 5). The dynamics with two stubborn agents show that consensus is formed locally among the tree module structures branching from the "main" link path connecting the two stubborn agents (a line network). The branching agent is a hub agent for its community. Consensus is formed among such local modules first. When there are no stubborn agents, a global convergence will follow.

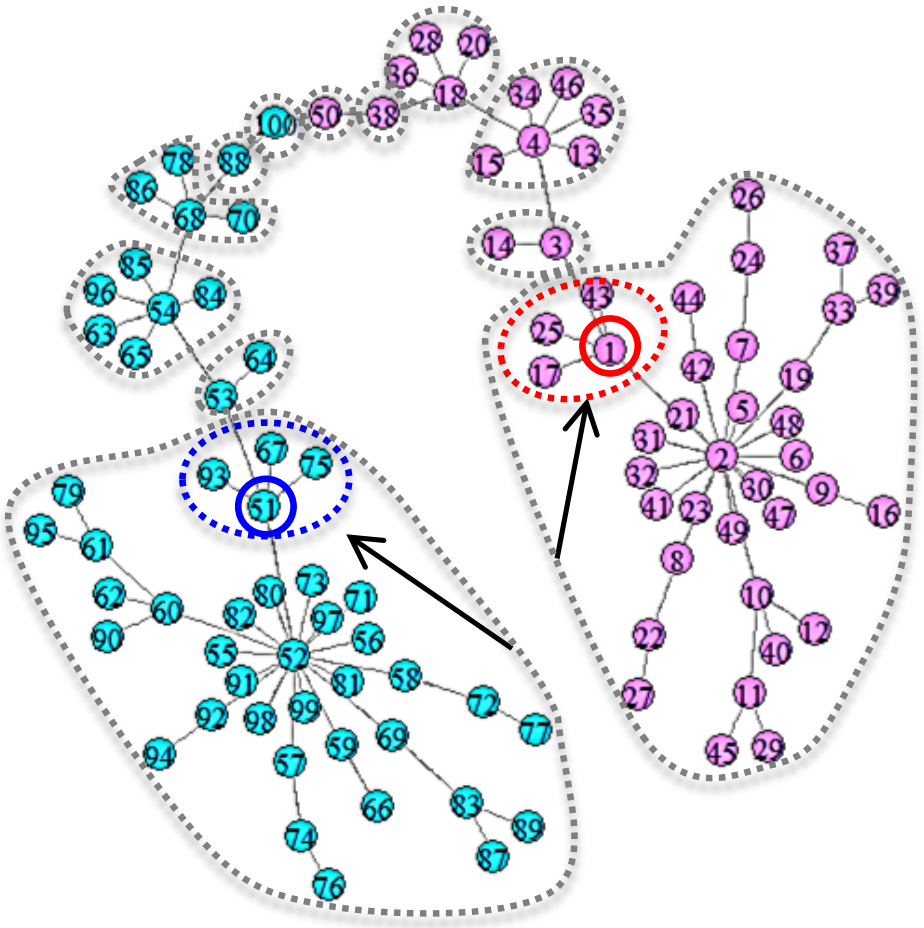


Fig. 4. Two scale-free networks connected by the most "remote" nodes. The agents forming local consensus are grouped by the gray dotted line. The stubborn agents are marked by the red and blue circles. The red and blue dotted lines are the local consensus initially formed by the tree module community with the stubborn agent as the hub. The black arrow shows that the larger consensus community eventually forms a consensus with the stubborn agent community, a dynamics which can also be observed in Fig. 5

However, when there are two stubborn agents located at the opposite ends of the line network, local consensus never merges together. In fact, the values taken by the local module consensus are essentially the hub agent number evenly dividing the agent value range (Fig. 5), a linear interpolation indicated by [25]. Consequently, the number of consensus settling values equals the number of hub agents along the main link path between the two stubborn agents, in addition to the two stubborn agent values. The order of consensus values associated with the hub communities is consistent with its order along the main link path that connects the two stubborn agents.

One consensus dynamics to notice is the convergence of two tree modules (Fig. 4). Initially, the tree modules of agents 1 and 2 form a consensus among their own modules. Then as time progresses, the consensus of these two modules converges. This occurs because agent 1 is a stubborn agent and all agents in the agent 2 module are regular agents. Therefore all agents in the agent 2 module form a consensus first. Then as time passes, they form a consensus with the stubborn agent 1. The result in the other half of the scale-free network follows a symmetric dynamics with agents 51 and 52.

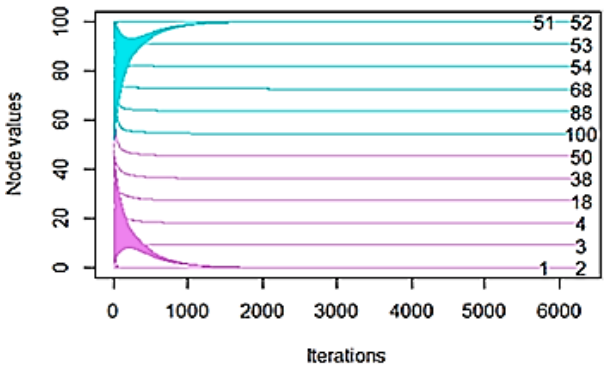


Fig. 5. Graph with the change in the agent value. *x*-axis is time step and *y*-axis is node value. The node numbers written on the agent values are the stubborn agents (1, 51) and the hub agents forming a consensus among their tree module along the main link path between the two stubborn agents

When two stubborn agents are located at the neighboring most "remote" linking agents at nodes 50 and 100, the two scale-free networks form a consensus only within their network community, i.e., the regular agents are attracted to the stubborn agent in their own community.

This is analogous to case 5 in the dumbbell topology dynamics. The value of consensus of each team depends on the value of their respective stubborn agent. Due to the limited node degree of these stubborn agents (agents 50 and 100), the consensus takes almost 10 times longer (about 59,000 time steps) than when the stubborn agents are placed at the nodes with the highest alpha-centrality (agents 1 and 51).

IV. DISCUSSION

This paper investigates the behavior of agents under the influence of opposite opinions or conflicting signals. The agents are caught between dynamics that aggregate and dynamics that are biased. When working in a team, the members must cope with both internal and external noise. Additionally, such local dynamics are met with the global constraint of a network. In the context of team negotiations in business or social settings, an analyst needs to take into account the impact of the social and organizational environments within which agent teamwork dynamics are embedded [38]. The agents' roles, the team characteristics, and agent relationship networks are critical sources of influence within a negotiations setting in particular [39], but this can be applied to other teamwork dynamics as well.

A. Minimum dumbbell team structure

The dumbbell network settings can be used to model a negotiation or a collaborative situation between two teams of agents on a single project. In case 1, a leader with a clear objective, vision, and resource can guide the two teams and the rest will follow (Fig. 3(a)). In case 2, two of the members of the same team may have opposite opinions (two stubborn agents), which is an intra-group conflict situation [40] (Fig. 3(b)). In a real world setting, the teammates are 'not on the same page,' 'confused about goals,' and possess 'conflicting interest within the team' [26]. This may happen when the agents represent different departments and have different goals to achieve. Since the members in the teams must maintain their respective teams (maintain the clique structure), the rest of the agents in both teams take an intermediate opinion and collaborate or find a common ground (form a local consensus) (Fig. 3(b), agents 1, 4, 5, 6). The solution to such intra-team conflict is to have prior preparation for a common agenda [26].

In case 3, the team with stubborn agents will be divided in 3 opinion values, the two stubborn values and the intermediate value of the agent that is connected to the two strong opposite opinions (Fig. 3(c)). If one of the bridge members has a strong vision and leadership (agent 1) and if all the agents in the opposite team are regular agents (agents 4, 5, 6), the opposite team will form an agreement with the bridging member of the team with stubborn agents (agent 1). The bridge agent is known as a boundary role person (BRP). Because such BRPs are often closer to the opposite team, physically and psychologically, they naturally share a relationship with the BRPs of the opposite team. This can potentially weaken the bond of the BRP with their own team [32]. After all, it is often preferable that negotiators have an understanding of and act in a way to accommodate the opponent's needs [31]. The influence of the other team is stronger with individualistic cultures in the occident than in the collectivist cultures in the orient, because BRP of the individualistic cultures are less bound to their own team [26].

In case 4, one of the bridging members, a BRP, is a stubborn agent or a leader (Fig. 3(d), agent 1). However, there is a strong opposing opinion in the other team which is not a BRP (agent 5). The members of the BRP leader team follow the BRP leader. Although the BRP leader's opinion has an influence on the team members of the other team, where the strongest influence is on the BRP of the other team and less influence on the non-BRP regular agents, the BRP leader cannot completely win over all the agents in the other team due to the opposition member (agent 5) in the other team. This case is a good example to show the extent of influence an agent would have depending on the location in a global network topology. One stubborn agent is a BRP (agent 1) and the other stubborn agent (agent 1) is not. In a team negotiation context, it is known that the quality of the negotiation outcome may be in favor of the BRP that is closer in psychological distance to the core ideologies (stubborn agent) of its own team. Perry [33] has concluded that when one of the negotiators is psychologically distant from their own team, the bargaining outcomes should fall in between the outcomes obtained when the negotiators are either close or far away psychologically. This outcome can be seen in the time progression of Fig. 3(d). Additionally, there is a profound significance to this in a corporate setting. As mentioned earlier, a BRP is located in a privileged position in that it is the only agent to reach the opposite team. Therefore, BRP is an agent that has access to internal as well as external (other corporate) information. Assume that BRP initially does not have any hierarchical authority or a legitimate basis of power. It is just another employee. However, during the time of environmental turbulence for business, due to its unique position in the interaction network of having both internal and external information, the BRP has the potential ability to cope with and absorb uncertainty, and channel or control the flow of information to its own company [41]. Since "power" is defined as an inverse of dependency [42], a BRP could gain power which initially would not have had if it were not a BRP. In fact, the BRP could intentionally generate conflict to maintain the company's dependency on itself and thus keep its power. This is seen again later in this paper.

In case 5, the connecting members of the two teams have opposite opinions (Fig. 3(e), agents 1 and 4). Since the regular agent members (agents 2, 3, 5, 6) of their respective teams are flexible and obedient to their strong leader, the two teams would not reach a consensus and therefore could not collaborate. This is a situation where the intensity of conflict is high. If a consensus of two teams must be achieved, the two teams may need a mediator or an arbitrator [43]. Mediation is where a third party makes a recommendation that is mutually acceptable based on information about the two teams. Arbitration is where the two teams commit to following the third party recommendation. The arbitrator has enforcing means. Both mediators and arbitrators are a third party that intervenes in the negotiation process [44]. In case 6, when there are oppositions by non-bridge members in each team (Fig. 3(f), agents 2 and 5), even though the members are attracted by the other team, they would not form a single consensus. Rather, the

opinion of the members would range between the two extreme opinions, where the intermediate BRPs would receive the greatest influence from the other team. Although in case 5 a consensus was formed in their respective cliques, the formation of coalitions is beginning to appear in case 4 and 6. This is more so when we have a tree module line network, which is discussed below.

B. Scale-free network team structure

So far, a model of interaction of small scale teams was considered. The location of stubborn agents and the role of the BRP were examined. When the number of agents increases in a team, there are two things that can change from a smaller group. The first is, as the number of agents increase, there will be more opinions and cultures. The second is, the BRP at the boundary surface may become more distant from the core ideology of their team held by a strong leader (stubborn agent).

First considering the cultural aspect, when the number of cultural diversity increases, the team is exposed to the possibility of forming coalitions within the team. Coalition is a subunit of a team that is consisted of two or more members whose intentions are to fulfill their own goals instead of the goals of the host team [45]. Coalition is said to form among those who have similar social/cultural traits. Here, social/cultural traits are individualism/collectivism, masculinity/femininity, etc. [46]. When the team is culturally homogeneous or completely heterogeneous, the likelihood of coalition formation is less. However, when the diversity of cultures is intermediate, a team is exposed to the possibility of coalition formation [47]. The presence of a coalition in a team can be unproductive because it may cause internal conflict and distancing among members [48]. Coalition makes the team more complex and unstable.

Assuming various cultures exist in a team, such cultures may be modeled as a sub-modular structure in the overall network, as seen in the hub and branch module structure in Fig. 4. When there are two stubborn agents, the agents and their cultural groups positioned between the two stubborn agents may hold varying degrees of sentiments depending on their location. During the consensus procedure, these network modules form a consensus within their cultural module. If there are no stubborn agents or if there is a single stubborn agent, these modules form a global consensus after they have established a local consensus as we have seen in Fig. 2. This two stage convergence is seen also for agents not in between, but "outside" the two stubborn agents, such as agents 2 and 52 community who eventually form a consensus with the agents 1 and 51 community. However, when there are two opposing state values (agent 1 and 51), the local culture modules between the two stubborn agents never converge with the other culture modules due to the influence of the opposing stubborn agents. Thus, multiple coalitions are formed within the team.

Finally, considering the distance of the BRPs from the core ideologies of their team, the farther away they are from the core and closer to the periphery near their opposite team, there is the risk of the negotiation outcome become less attractive for its own team where the BRP belongs and more preferable for the opposite team. There are two reasons why this may happen. First, because of the distance of the BRP from the core, the negotiator BRP may have an ambiguous understanding of the team's goal because of conflicting information. The second reason is the BRP may set incongruent goals of their team, either unintentionally due to lack of information or intentionally in order to create uncertainty in the situation so that they would be relied upon more by their organization and thus gaining more power. The farther away the BRP is from the core, this trend is stronger [33]. The terminal values taken by the BRP agents in Fig. 5 reflect this situation. The closer the BRP is to their opposite team and farther away they are from their respective "core" stubborn agent values, the BRP values of both teams become closer.

V. Conclusion and Summary

A multi-agent system was used to model the interaction of agents under the restrain of a team structure. The agents have an averaging protocol for consensus formation. However, not all agents are cooperative. The team interaction structures used were a minimum dumbbell network and two scale-free networks connected. All possible outcomes of the minimum dumbbell structure with two stubborn agents were interpreted in a team cooperation/negotiation setting. As indicated by previous literature, the bridging agents or boundary role persons have a unique role to serve. Its position on the network showed that it has a strong influence over the other agents. When stubborn agents are located at both bridging nodes, the agents are strongly united with their team that it is recommended to involve a third party to moderate or arbitrate.

The scale-free line network with tree modules showed the possibility of coalition formation based on the cultural diversity in the team. Such coalition formation dynamics can be seen in the way modular networks form a consensus, that local consensus is formed first and then a global consensus is reached. This shows the influence of both local and global topology that influences the overall characteristics of the network dynamics. The branching agents from the main line nodes can be considered to have the same culture/attitude/mentality. This network can also model the psychological distance between the boundary role person and the core ideology of the team. The farther away they are from the core, and more cultural values there are in between, their ideology becomes closer to the other team. Such a model can be used not only for commercial settings, but also in a political setting in society, between two extreme policies.

References

[1] T. H. Cox, Cultural Diversity in Organizations: Theory, Research and Practice. San Francisco, CA: Barrett-Koehler, 1993.

[2] S. Jackson, K. E. May, and K. Whitney, "Understanding the dynamics of diversity in decisions making teams," in Team Effectiveness and Decision Making in Organizations, R. A. Guzzo and E. Salas, Eds. San Francisco, CA: Jossey-Bass, 1995, pp. 204–261.

[3] D. R. Illgen, J. R. Hollenbeck, M. Johnson, and D. Jundt, "Teams in organizations: From I-P-O models to IMOI models," Annual Review, vol. 56, pp. 517–543, 2006.

[4] A. W. Richter, M. A. West, R. van Dick, and J. F. Dawson, "Boundary spanners' identification, intergroup contact, and effective intergroup relations," The Academy of Management Journal, vol. 49, pp. 1252–1269, 2006.

[5] P. L. Curşeu, P. Kenis, J. Raab, and U. Brandes, "Composing effective teams through team-dating," Organ. Stud., vol. 31, pp. 873–894, 2010.

[6] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," Nature, vol. 393, pp. 440–442, 1998.

[7] A. L. Barabási, Linked: The New Science of Networks, Cambridge, MA: Perseus Publishing, 2002.

[8] A. Pikovsky, M. Rosenblum, and J. Kurths, Synchronization: A Universal Concept in Nonlinear Sciences, Cambridge, UK: Cambridge University presses, 2003.

[9] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems", in proc. of IEEE, vol. 95, no. 1, pp.215–233, 2007.

[10] M. Barahona, "Synchronization in small-world systems," Physical Review Letters, vol. 89, 2002.

[11] X. F. Wang and G. Chen, "Synchronization in scale-free dynamical networks: Robustness and fragility", IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, pp.54–62, 2001.

[12] V. H. P. Louzada, N. A. Araujo, and H. Herrmann, "How to suppress undesired synchronization," Scientific Reports 2, no. 658, 2012.

[13] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, "Opinion fluctuations and disagreement in social networks," Mathematics of Operation Research, vol. 38, no. 1, pp. 1–27, 2013.

[14] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," IEEE Transactions on Automatic Control, vol. 53, no. 8, pp. 1972–1975, 2008.

[15] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1520–1533, 2004.

[16] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," IEEE Transactions on Robotics and Automation, vol. 20, no. 2, pp. 243–255, 2004.

[17] V. Gupta, C. Langbort, and R. Murray, "On the robustness of distributed algorithms," in proc. of the 45th IEEE Conference on Decision and Control, 2006.

[18] N. Lynch, Distributed Algorithms. San Mateo, CA: Morgan Kaufmann Publishers, 1996.

[19] F. Fagnani, "Consensus dynamics over networks," 2014.

[20] L. Lamport, R. Shostak and M. Pease, "The Byzantine generals problem,' ACM Transactions on Programming Languages and Systems, vol. 4, no. 3, pp. 382–401, 1982.

[21] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi, "Spread of (mis)information in social networks," Games and Economic Behavior, vol. 70, no. 2, pp. 194–227, 2010.

[22] E. Yildiz , D. Acemoglu, A. Ozdaglar, A. Saberi, and A. Scaglione, "Discrete opinion dynamics with stubborn agents," 2011.

[23] R. A. Holley and T. M. Liggett, "Ergodic theorems for weakly interacting infinite systems and the voter model," The Annals of Probability vol. 3, no. 4, pp. 643–663, 1975.

[24] P. Clifford and A. Sudbury, "A model for spatial conflict," Biometrika, vol. 60, no. 3, pp. 581–588, 1973.

[25] D. Acemoglu, G. Como, F. Fagnani, and A. E. Ozdaglar, "Opinion fluctuations and disagreement in social networks," Math. Oper. Res. vol. 38, no.1, pp. 1–27, 2013.

[26] K. Behfar, R. Friedman, and J. M. Brett, "The team negotiation challenge: defining and managing the internal challenges of negotiating teams," IACM 21st Annual Conference Paper, November, 2008.

[27] D. S. Alberts and R. E. Hayes, Power to the edge: Command…control…in the information age. Washington: CCRP, 2003.

[28] D. S. Alberts, "Agility, Focus, and Convergence: The Future of Command and Control," The International C2 Journal, vol. 1, pp.1–30, 2007.

[29] S. Brodt and L. Thompson, "Negotiating teams: A levels of analysis approach," Group Dynamics-Theory Research and Practice, vol. 5, no. 3, pp. 208–219, 2001.

[30] M. Barthélemy, "Betweenness centrality in large complex networks," The European Physical Journal B - Condensed Matter and Complex Systems, vol. 38, issue 2, pp. 163–168, 2004.

[31] R. E. Walton and R. B. McKersie, A Behavioral Theory of Labor Negotiations: An Analysis of a Social Interaction System. New York: McGraw-Hill, 1965.

[32] J. S. Adams, "The structure and dynamics of behavior in organization boundary roles," in Handbook of Industrial and Organizational Psychology, M. D. Dunnette, Ed. Chicago: Rand McNally, 1976, pp. 1175–1199.

[33] J. L. Perry and H. L. Angle, "The politics of organizational boundary roles in collective bargaining," The Academy of Management Review, vol. 4, no. 4, pp. 487–495, October 1979.

[34] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," Systems & Control Letters, vol. 53, no.1, pp. 65–78, 2004.

[35] W. Ren, R. W. Beard, and E. M. Atkins, "A Survey of Consensus Problems in Multi-agent Coordination," in proc. of the American Control Conference, 2005.

[36] L. Thompson, E. Peterson, and S. Brodt, "Team negotiation: an examination of integrative and distributive bargaining," J. Per. Soc. Psychol., vol. 70, pp. 66–78, 1996.

[37] P. Bonacich and P. Lloyd, "Eigenvector-like measures of centrality for asymmetric relations," Social Networks, vol. 23, pp. 191–201, 2001.

[38] R. M. Kramer and D. M. Messick, Negotiation as a Social Process. Thousand Oaks, CA: Sage, 1995.

[39] M. J. Gelfand and J. M. Brett, The Handbook of Negotiation and Culture. Stanford, CA: Stanford University Press, 2004.

[40] K. Jehn, "A qualitative analysis of conflict types and dimensions in organizational groups," Administrative Science Quarterly, vol. 42, pp. 530–557, 1997.

[41] R. E. Spekman, "Influence and information: An exploratory investigation of the boundary role person's basis of power," Academy of Management Journal, vol. 22, no. 1, 1979.

[42] R. Emerson, "Power dependence relations," American Sociological Review, vol. 22, pp. 31–41, 1962.

[43] M. Goltsman, J. Hörner, G. Pavlov, and F. Squintani, "Mediation, arbitration and negotiation," Journal of Economic Theory, vol. 144, no. 4, pp. 1397–1420, 2009.

[44] R. J. Lewicki, S. E. Weiss, and D. Lewin, "Models of conflict, negotiation and third party intervention: A review and synthesis," Journal of Organizational Behavior, vol. 13, no. 3, pp. 209–252, May 1992.

[45] J. K. Murnighan, "Organizational coalitions: structural contingencies and the formation process," in Research on Negotiation in Organizations, vol. 1, R. J. Lewicki, B. H. Sheppard, and M. Bazerman, Eds. Stanford, CT: JAI Publishing, 1986, pp. 155–173.

[46] G. Hofstede, Culture's Consequences, International Differences in Work-related Values, Beverly Hills, CA: Sage, 1980.

[47] X. Guo and J. Lim, "Negotiation support systems and team negotiations: The coalition formation perspective," Information and Software Technology, vol. 49, pp. 1121–1127, 2007.

[48] S. M. B. Thatcher, K. A. Jehn, and E. Zanutto, "Cracks in diversity research: The effects of diversity faultlines on conflict and performance," Group Decis. Negotiation, vol. 12, pp. 217–241, 2003

Integral Lqr-Based 6dof Autonomous Quadrocopter Balancing System Control

A Joukhadar, BSc, MPhil, PhD
Dept. of Mechatronics Engineering
University of Aleppo, Aleppo-Syria

A Alsabbagh, BSc(Candidate)
Dept. of Mechatronics Engineering
University of Aleppo, Aleppo-Syria

I Hasan, BSc(Candidate)
Dept. of Mechatronics Engineering
University of Aleppo, Aleppo-Syria

M Alkouzbary, BSc(Candidate)
Dept. of Mechatronics Engineering
University of Aleppo, Aleppo-Syria

***Abstract*—This paper presents an LQR-Based 6DOF control of an unmanned aerial vehicles (UAV), namely a small-scale quadrocopter. Due to its high nonlinearity and a high degree of coupling system, the control of an UAV is very challenging. quadrocopter trajectory tracking in a 3D space is greatly affected by the quadrocopter balancing around its roll-pitch-yaw frame. Lack of precise tracking control about the body frame may result in inaccurate localization with respect to a fixed frame. Thus, the present paper provides a high dynamic control tracking balancing system response. An integral LQR-based controller is proposed to enhance the dynamic system response balancing on roll, pitch and yaw. The control on the hovering angles consists of two-cascaded loops. Namely, an inner loop for the angular speed control of each angular motion around the body frame axes, and an outer loop for the desired position control. In general, the proposed balancing control system on roll, pitch and yaw, has six control loops. The proposed control approach is implemented utilizing an embedded ATMega2560 microcontroller system. Practical results obtained from the proposed control approach exhibits fast and robust control response and high disturbance rejection.**

Keywords—Quadrocopter; Balancing Control; Stability of Quadrocopter; LQR; Integral LQR; Modelling of Quadrocopter

I. INTRODUCTION

Control of Unmanned Aerial Vehicles, known as UAVs, has been considered as one of the difficult and complicated challenges, especially controlling those that can perform vertical take-off and landing (VTOL). quadrocopters have become very popular recently due to their simple structure design compared with what they can perform even in a complex environment. Moreover, recent development in high density power storage, integrated miniature actuators and MEMS technology sensors have made the autonomous flying robots possible. However, the quadrocopter is a very nonlinear system; that makes the control of this vehicle not easy to be accomplished.

The movement of the vehicle's body frame with respect to the inertial earth frame is controlled by adjusting the angular speed of the propellers properly. Control laws will lead the vehicle to act in a particular manner as desired. Several control methods have been tested on the quadrocopter to stabilize and track control.

In [1] & [2], authors compared between PID, Sliding mode, LQR and Backstepping, for attitude control only for the quadrocopter. Authors in [1], propose an Integral Backstepping for full position control of the quadrocopter system. LQG controller is applied to control the attitude of quadrocopter [3].

Feedback linearization based technique is proposed to deduce the control law for quadrocopter attitude control, [4]. In [5], a modified Backstepping approach is proposed to control attitude and position of the quadrocopter, where the main contribution was to reduce the number of gains in the control laws.

In [6], a nonlinear control design is combined with an online parameter estimation to develop the control law, in presence of parameter uncertainties and compared with sliding mode control. In [7], Fuzzy Backstepping Sliding Mode Controllers are designed for quadrocopter. However, this control technique is a Backstepping technique where the error signals were determined as the sliding manifold, moreover, a Fuzzy Controler was added instead of sign function in the control law.

The remaining sections are, section II, which briefs the quadrocopter system description. The quadrocopter LQR Controller is discussed in section III. The dynamic model of quadrocopter is given in section IV. Section V explains the design of LQR optimal control design. Practical implementation and results are given in section VI. Section VII discusses the concluded practical results.

TABLE I. SYMBOLS TABLE

Symbol	Units	Brief description
F_E	*None*	Earth frame
F_B	*None*	Body frame
$\boldsymbol{\xi}$	*m*	Position vector in the earth frame
X	*m*	x unit vector in the earth frame
Y	*m*	y unit vector in the earth frame
Z	*m*	z unit vector in the earth frame
$\boldsymbol{\eta}$	*deg*	Orientation angle vector in the earth frame
ϕ	*deg*	Roll Euler angle
θ	*deg*	Pitch Euler angle
ψ	*deg*	Yaw Euler angle
$\mathbf{V}_B$	*m/s*	The prismatic speed vector in the body frame
u	*m/s*	The prismatic speed on x unit vector in the body frame
v	*m/s*	The prismatic speed on y unit vector in the body frame
w	*m/s*	The prismatic speed on z unit vector in the body frame
v	*deg/s*	The angular speed vector in the body frame
p	*deg/s*	The angular speed on x unit vector in the body frame
q	*deg/s*	The angular speed on y unit vector in the body frame
r	*deg/s*	The angular speed on z unit vector in the body frame
R	*None*	The transformation matrix from the body frame to the earth frame
$\mathbf{R}^{-1}$	*None*	The transformation matrix from the earth frame to the body frame
$\mathbf{W}_{\eta}^{-1}$	*None*	The transformation matrix of angular speed from the body frame to the earth frame
m	*kg*	Vehicle flight mass
$\mathbf{T}_B$	*m/s*	Vector of total force acting on vehicle expressed in the body frame
G	*N.m*	Vector force due to gravity acting on vehicle expressed in the body frame
I	$kg.m^2$	Vehicle flight moment of inertia tensor w.r.t. Center of mass
I_{xx}	$kg.m^2$	x principle moment of inertia
I_{yy}	$kg.m^2$	y principle moment of inertia
I_{zz}	$kg.m^2$	z principle moment of inertia
f_i	*N*	Vector thrust of rotor i expressed in the body fram
b	$N.s^2$	Thrust factor
d	$N.m.s^2$	Drag factor
Ω_i	*deg/s*	Scalar rotational speed of rotor *i*
I_M	$kg.m^2$	Motor moment of inertia
L	*m*	Horizontal distance: propeller center to cog.
$\boldsymbol{\tau}$	*N.m*	Vector of total torque acting on vehicle expressed in the body frame
$\boldsymbol{\Gamma}$	*N.m*	Gyroscopic force vector
J_r	$kg.m^2$	Rotor moment of inertia
Ω_d	*deg/s*	Overall residual propeller angular speed

II. QUADROCOPTER SYSTEM DESCRIPTION

The UAV system used for implementation, is shown in Fig. 2. It is aimed to make an autonomous quadrocopter, which can balance itself while flying. To have it done, an Inertial Measurement Unit (IMU) and an altimeter (altitude sensor) are used. The IMU is an MEMS type that measures and reports the quadrocopter's orientation and gravitational forces, using a combination of accelerometer, gyroscope and magnetometer. The altimeter is a fusion sensor that measures the altitude of the quadrocopter. Four BLDC motors, out-runner type, are used. Fig. 2 shows an image of the quadrocopter under investigation in the Lab.

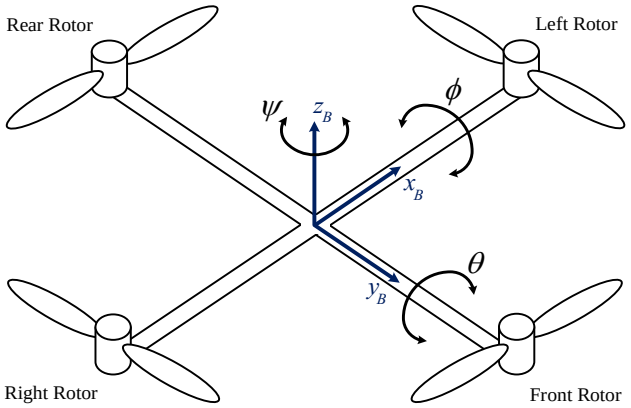


Fig. 1. Euler angles

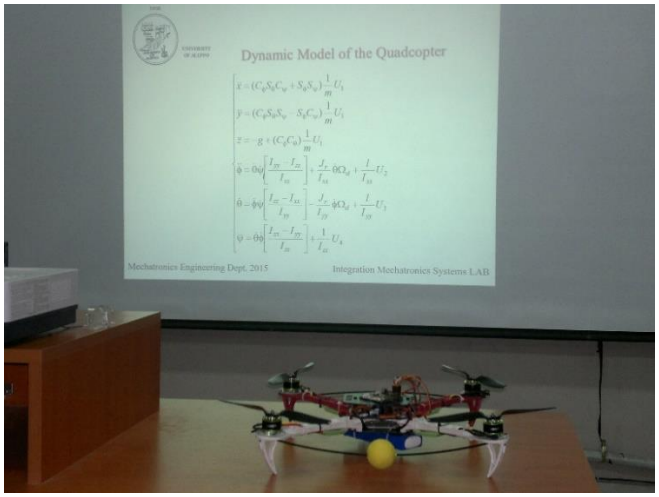


Fig. 2. Quadrocopter system under investigation

The quadrocopter is equipped with an embedded fight controller board, which consists of an ATMega2560 microcontroller and MPU6050 integrated 6-axis motion tracking device, which combines 3-axis gyroscope and 3-axis accelerometer with its dedicated I2C sensor bus. It directly accepts inputs from an external 3-axis compass to provide a complete 9-axis Motion Fusion. Fig. 3 shows the flight controller board. The board also consists of HMC5883L 3-axis digital magnetometer

is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. MS5611-01BA03 is a new generation of high-resolution altimeter sensors from MEAS Switzerland with SPI and I2C bus interface.

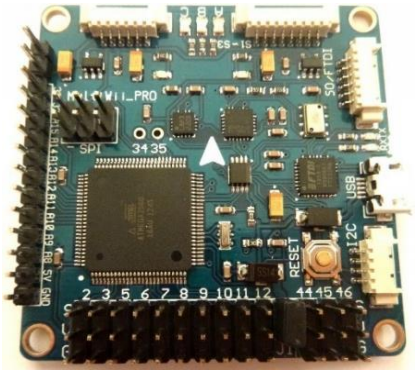


Fig. 3. Mircontroller board

III. Proposed Quadrocopter LQR Control

Fig. 4 shows the proposed entire control system for qaudrocopter balancing and localization. The main goal of the paper is to develop an integral LQR controller to enhance the dynamic control response of the quadrocopter with respect to the quadrocopter body frame. As seen from Fig. 4, the proposed control system consists of two control levels. This includes a low-level control (body frame control), and a high-level control.

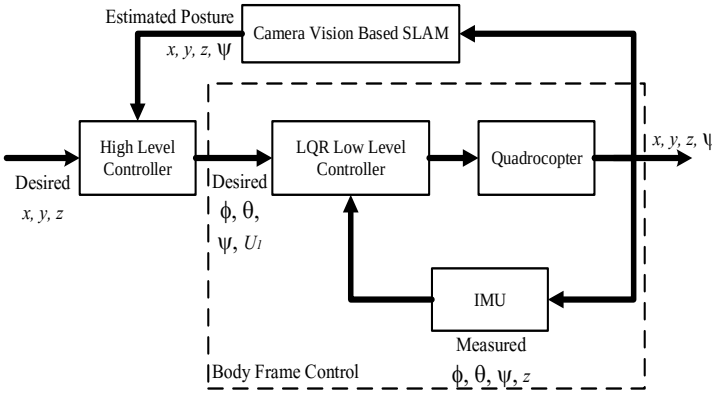


Fig. 4. Proposed Quadrocopter Control System approach

The main task of the low level control, is to improve the dynamic performance of the quadrocopter on the body frame F_B. The low-level control receives the desired command signals, ϕ and θ from the high-level control. While, the control law for the orientation angle ψ considered a high-level control. It is wise mentioning that the goal of the high-level control is to precisely localize the quadrocopter at a desired posture with respect to a fixed earth frame F_E [3].

IV. Dynamic Model of a Quadrocopter

The non-linear dynamic model of the quadrocopter is provided [8]. It is of 6DOF system, which includes, the position vector $\boldsymbol{\xi}$ with respect to a fixed frame and the orientation vector $\boldsymbol{\eta}$ of the body frame F_B relative to a fixed origin of the earth frame F_E. Fig. 5 shows the quadrocopter system with its body frame referenced to a fixed frame [9] & [10].

$$\boldsymbol{\xi} = [x \;\; y \;\; z]^{\mathrm{T}} \tag{1}$$

$$\boldsymbol{\eta} = [\phi \;\; \theta \;\; \psi]^{\mathrm{T}} \tag{2}$$

$$\mathbf{q} = [\boldsymbol{\xi} \;\; \boldsymbol{\eta}]^{\mathrm{T}} \tag{3}$$

Equation (3) represents the joint system state space vector of the 6DOF quadrocopter system, which includes the position and the orientation vectors (1) and (2). The linear and the angular velocity vectors are given by (4) and (5), respectively.

$$\mathbf{V}_B = [u \;\; v \;\; w]^{\mathrm{T}} \tag{4}$$

$$\boldsymbol{\upsilon} = [p \;\; q \;\; r]^{\mathrm{T}} \tag{5}$$

The rotation matrix $\mathbf{R}$ from body frame F_B to earth frame F_E, and the transformation matrix for angular speeds from F_B to F_E, are given by (6) and (7), respectively.

$$\mathbf{R} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{6}$$

$$\mathbf{W}_\eta^{-1} = \begin{bmatrix} 1 & S_\phi T_\theta & C_\phi T_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \tag{7}$$

In which, $S_\phi = \sin\phi, C_\phi = \cos\phi, T_\phi = \tan\phi$

And the rotation matrix is orthogonal, i.e., $\mathbf{R}^{-1}=\mathbf{R}^{\mathrm{T}}$. The physical structure of the quadrocopter is symmetrical about all axes; hence, the inertial matrix is defined as in (8).

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{8}$$

When rotor i rotates, it generates a lift force f_i, which causes a vertical motion, and an angular torque i around z-axis τ_{Mi} as given in (9) and (10).

$$f_i = b\Omega_i^2 \tag{9}$$

$$\tau_{Mi} = d\Omega_i^2 + I_M\Omega_i \tag{10}$$

As a result, three forces and one torque affect the quadrocopter body and are determined as given by (11), (12), (13) and (14).

$$U_1 = \sum_{i=1}^{4} f_i = b\sum_{i=1}^{4}\Omega_i^2 \tag{11}$$

$$U_2 = b(-\Omega_2^2 + \Omega_4^2) \tag{12}$$

$$U_3 = b(-\Omega_1^2 + \Omega_3^2) \tag{13}$$

$$U_4 = \sum_{i=1}^{4}\tau_{Mi} \tag{14}$$

$$\mathbf{T}_B = [0 \;\; 0 \;\; U_1]^{\mathrm{T}} \tag{15}$$

$$\boldsymbol{\tau} = [lU_2 \;\; lU_3 \;\; U_4]^{\mathrm{T}} \tag{16}$$

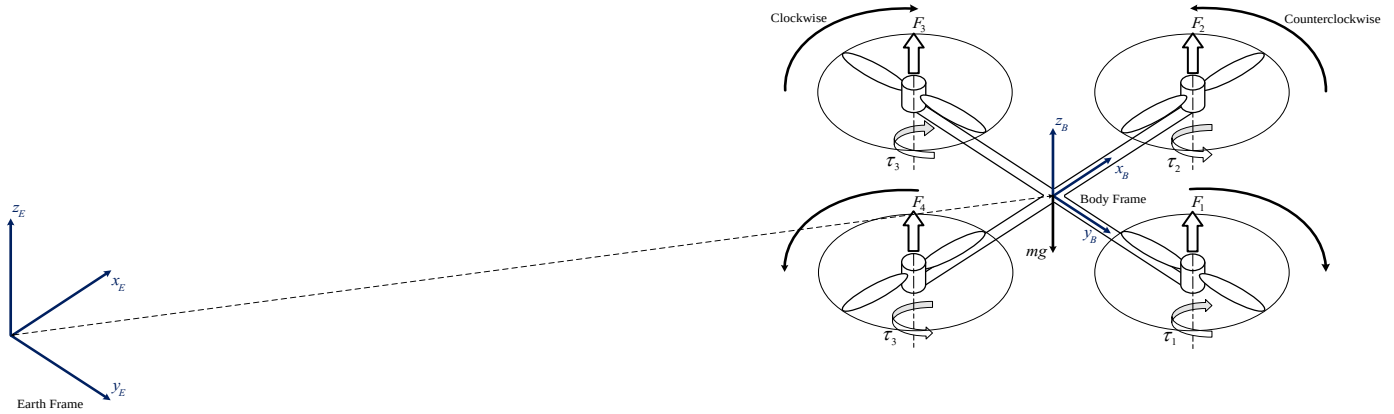


Fig. 5. Reference frames F_B and F_E, and forces and torques generated by the rotors of the quadrocopter

A. Newton-Euler Equations

It is assumed the quadrocopter to be a rigid body; hence, Newton-Euler equations can be used to describe its dynamics. In the body frame F_B, the force required for the acceleration of mass $m\dot{\mathbf{V}}_B$, and the centrifugal force $\boldsymbol{\upsilon}\times(m\mathbf{V}_B)$ are equal to gravity $\mathbf{R}^{\mathrm{T}}\mathbf{G}$ and the total thrust of the rotors $\mathbf{T}_B$, which is defined in (15), the outcome equation is seen in (18).

$$m\dot{\mathbf{V}}_B+\boldsymbol{\upsilon}\times(m\mathbf{V}_B)=\mathbf{R}^{\mathrm{T}}\mathbf{G}+\mathbf{T}_B \tag{18}$$

In F_E, the centrifugal force is nullified. Therefore, only the gravitational force and the magnitude and direction of the thrust are contributing in the acceleration of the quadrocopter, as in (19) and (20).

$$m\ddot{\boldsymbol{\xi}}=\mathbf{G}+\mathbf{R}\mathbf{T}_B \tag{19}$$

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}=\begin{bmatrix}0\\0\\-g\end{bmatrix}+\frac{U_1}{m}\begin{bmatrix}C_\psi S_\theta C_\phi+S_\psi S_\phi\\ S_\psi S_\theta C_\phi-C_\psi S_\phi\\ C_\theta C_\phi\end{bmatrix} \tag{20}$$

In F_B, the angular acceleration of the inertia $\mathbf{I}\dot{\boldsymbol{\upsilon}}$, the centripetal forces $\boldsymbol{\upsilon}\times(\mathbf{I}\boldsymbol{\upsilon})$ and the gyroscopic forces $\boldsymbol{\Gamma}$ are equal to the external torque $\boldsymbol{\tau}$, as in (21), (22) and (23).

$$\boldsymbol{\tau}=\mathbf{I}\dot{\boldsymbol{\upsilon}}+\boldsymbol{\upsilon}\times(\mathbf{I}\boldsymbol{\upsilon})+\boldsymbol{\Gamma} \tag{21}$$

$$\boldsymbol{\upsilon}\times(\mathbf{I}\boldsymbol{\upsilon})=\begin{bmatrix}p\\q\\r\end{bmatrix}\times\begin{bmatrix}I_{xx}p\\I_{yy}q\\I_{zz}r\end{bmatrix}=\begin{bmatrix}(I_{zz}-I_{yy})qr\\(I_{xx}-I_{zz})pr\\(I_{yy}-I_{xx})pq\end{bmatrix} \tag{22}$$

$$\boldsymbol{\Gamma}=J_r\begin{bmatrix}p\\q\\r\end{bmatrix}\times\begin{bmatrix}0\\0\\\Omega_d\end{bmatrix}=J_r\begin{bmatrix}q\Omega_d\\-p\Omega_d\\0\end{bmatrix} \tag{23}$$

In which,

$$\Omega_d=-\Omega_1+\Omega_2-\Omega_3+\Omega_4 \tag{24}$$

Consequently, the resulted equation can be written as in (25).

$$\begin{bmatrix}\dot{p}\\\dot{q}\\\dot{r}\end{bmatrix}=\mathbf{I}^{-1}\left\{\begin{bmatrix}lU_2\\lU_3\\U_4\end{bmatrix}-\begin{bmatrix}(I_{zz}-I_{yy})qr\\(I_{xx}-I_{zz})pr\\(I_{yy}-I_{xx})pq\end{bmatrix}-J_r\begin{bmatrix}-q\Omega_d\\p\Omega_d\\0\end{bmatrix}\right\} \tag{25}$$

$$\begin{bmatrix}\dot{p}\\\dot{q}\\\dot{r}\end{bmatrix}=\begin{bmatrix}lU_2/I_{xx}\\lU_3/I_{yy}\\U_4/I_{zz}\end{bmatrix}-\begin{bmatrix}(I_{zz}-I_{yy})qr/I_{xx}\\(I_{xx}-I_{zz})pr/I_{yy}\\(I_{yy}-I_{xx})pq/I_{zz}\end{bmatrix}-J_r\begin{bmatrix}-q\Omega_d/I_{xx}\\p\Omega_d/I_{yy}\\0\end{bmatrix} \tag{26}$$

Whereas,

$$\begin{bmatrix}\dot{\phi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix}=\begin{bmatrix}1&S_\phi T_\theta&C_\phi T_\theta\\0&C_\phi&-S_\phi\\0&S_\phi/C_\theta&C_\phi/C_\theta\end{bmatrix}\begin{bmatrix}p\\q\\r\end{bmatrix} \tag{27}$$

B. Nonlinear Dynamic Model Simplification

The transformation between $[\dot{p},\dot{q},\dot{r}]$ and $[\ddot{\phi},\ddot{\theta},\ddot{\psi}]$ for rotational dynamics, is very complex, since it includes many trigonometric functions; therefore, simplification is needed. It is assumed that if perturbations from hover condition are small, body angular rates and rate of change of Euler angles are equal for small values of ϕ and θ, the relation between body angular rates and rate of change of Euler angles becomes as in (28) [10].

$$\begin{bmatrix}\dot{\phi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix}=\begin{bmatrix}p\\q\\r\end{bmatrix},\begin{bmatrix}\ddot{\phi}\\\ddot{\theta}\\\ddot{\psi}\end{bmatrix}=\begin{bmatrix}\dot{p}\\\dot{q}\\\dot{r}\end{bmatrix} \tag{28}$$

According to previous, the complete dynamics of the vehicle is described in (29).

$$\begin{cases}\ddot{\phi}=a_1\dot{\theta}\dot{\psi}+b_1\dot{\theta}\Omega_d+c_1U_2\\ \ddot{\theta}=a_2\dot{\phi}\dot{\psi}-b_2\dot{\phi}\Omega_d+c_2U_3\\ \ddot{\psi}=a_3\dot{\theta}\dot{\phi}+c_3U_4\\ \ddot{x}=(C_\phi S_\theta C_\psi+S_\phi S_\psi)d_1U_1\\ \ddot{y}=(C_\phi S_\theta S_\psi-S_\phi C_\psi)d_1U_1\\ \ddot{z}=-g+(C_\phi C_\theta)d_1U_1\end{cases} \tag{29}$$

Whereas,

$a_1 = (I_{yy} - I_{zz})/I_{xx}, a_2 = (I_{zz} - I_{xx})/I_{yy}, a_3 = (I_{xx} - I_{yy})/I_{zz}$

$b_1 = J_r/I_{xx}, b_2 = J_r/I_{yy}$

$c_1 = l/I_{xx}, c_2 = l/I_{yy}, c_3 = l/I_{zz}$

$d_1 = 1/m$

V. LQR Optimal Controller Design

A. Dynamic Model Linearization

To achieve optimal control algorithm [11] & [12], Linear Quadratic Regular (LQR), the dynamic model, described in (29), must be linearized around a trim condition, which is chosen to be hover condition.

Defining $\mathbf{X} = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^{\mathrm{T}}$ as the state vector of the attitude dynamics whereas, $x_1 = \phi,\ x_2 = \dot{\phi},\ x_3 = \theta,\ x_4 = \dot{\theta},\ x_4 = \dot{\theta},\ x_5 = \psi,\ x_6 = \dot{\psi},$ and $\mathbf{U} = [U_1, U_2, U_3, U_4]^{\mathrm{T}}$ as the input vector. The state space representation of the dynamics can be given by $\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{U})$ where,

$$f(\mathbf{X}, \mathbf{U}) = \begin{bmatrix} x_2 \\ a_1 x_4 x_6 + b_1 x_4 \Omega_d + c_1 U_2 \\ x_4 \\ a_2 x_2 x_6 - b_2 x_2 \Omega_d + c_2 U_3 \\ x_6 \\ a_3 x_4 x_2 + c_3 U_4 \end{bmatrix} \tag{30}$$

To linearize the system given in (30), and from the Jacobian matrices described in (31) and (32), the linearized system will become as in (33).

$$\mathbf{A}_{n\times n} = \left[\frac{\partial f}{\partial \mathbf{X}}\right]_{(\mathbf{X}_0, \mathbf{U}_0)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{(\mathbf{X}_0, \mathbf{U}_0)} \tag{31}$$

$$\mathbf{B}_{n\times m} = \left[\frac{\partial f}{\partial \mathbf{U}}\right]_{(\mathbf{X}_0, \mathbf{U}_0)} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_{(\mathbf{X}_0, \mathbf{U}_0)} \tag{32}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ c_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c_3 \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{33}$$

To stabilize the quadrocopter, the system state, described in (33), will be divided into subsystems as follows:

$$\ddot{\phi} = c_1 U_2 \tag{34}$$

$$\dot{\theta} = c_2 U_3 \tag{35}$$

$$\ddot{\psi} = c_3 U_4 \tag{36}$$

This model could be optimally stabilized using the classical result of LQR.

B. Optimal stabilization of subsystem ϕ

First, the subsystem in (34) is written in a space state form as seen in (37).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ c_1 \end{bmatrix} U_2 \tag{37}$$

Fig. 6 shows the suggested LQR controller structure, where a feed forward loop has been added to obtain stabilization at steady state, which supports the control law to maintain the desired angle at its commanded value, and improve the system tracking response. The structure of this control regime was implemented to enhance the quadrocopter balancing system on both roll and pitch angles. It consists of LQR and Integral LQR type controllers. Different control approach structure was proposed for the yaw angle control. Fig. 7 explains the structure of the proposed LQR controller.

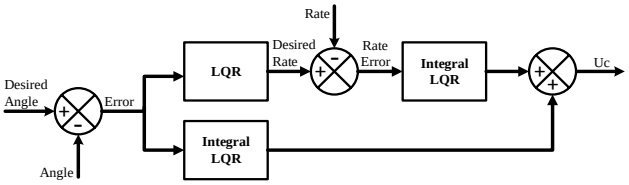


Fig. 6. Proposed optimal controller for roll and pitch angles control

The system given in (37) should be extended to apply the desired controller, and the extended system is shown in (38).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ c_1 \\ 0 \\ 0 \end{bmatrix} U_2 \tag{38}$$

$$\mathbf{Q}_\phi = \begin{bmatrix} 170 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 320 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}, \mathbf{R}_\phi = [100] \tag{39}$$

$$\mathbf{K}_\phi = [1.0333 \quad 0.447 \quad -1.791 \quad -0.7841] \tag{40}$$

C. Optimal stabilization of subsystem θ

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ c_2 \end{bmatrix} U_3 \tag{41}$$

$$\begin{bmatrix}\dot{x}_3\\ \dot{x}_4\\ \dot{\xi}_3\\ \dot{\xi}_4\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ -1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_3\\ x_4\\ \xi_3\\ \xi_4\end{bmatrix}+\begin{bmatrix}0\\ c_2\\ 0\\ 0\end{bmatrix}U_3 \quad (42)$$

$$\mathbf{Q}_\theta=\begin{bmatrix}170 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 320 & 0\\ 0 & 0 & 0 & 0.001\end{bmatrix},\mathbf{R}_\theta=\begin{bmatrix}100\end{bmatrix} \quad (43)$$

$$\mathbf{K}_\theta=\begin{bmatrix}1.0333 & 0.447 & -1.791 & -0.7841\end{bmatrix} \quad (44)$$

D. Optimal stabilization of subsystem ψ

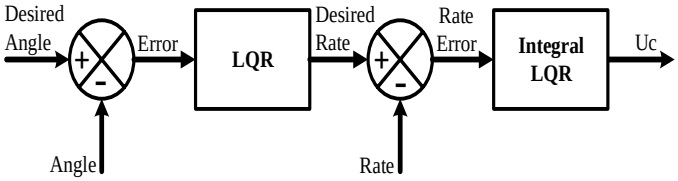


Fig. 7. The applied optimal controller to control yaw angle

$$\begin{bmatrix}\dot{x}_5\\ \dot{x}_6\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_5\\ x_6\end{bmatrix}+\begin{bmatrix}0\\ c_3\end{bmatrix}U_4 \quad (45)$$

$$\begin{bmatrix}\dot{x}_5\\ \dot{x}_6\\ \dot{\xi}_5\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & -1 & 0\end{bmatrix}\begin{bmatrix}x_5\\ x_6\\ \xi_5\end{bmatrix}+\begin{bmatrix}0\\ c_3\\ 0\end{bmatrix}U_4 \quad (46)$$

$$\mathbf{Q}_\psi=\begin{bmatrix}1 & 0 & 0\\ 0 & 1.1 & 0\\ 0 & 0 & 0.0001\end{bmatrix},\mathbf{R}_\psi=\begin{bmatrix}2.5\end{bmatrix} \quad (47)$$

$$\mathbf{K}_\psi=\begin{bmatrix}0.8421 & 0.6872 & -0.2067\end{bmatrix} \quad (48)$$

VI. Practical Implementation and Results

Practical results obtained from the test bench of the quadrocopter system, built in Lab, were demonstrated to validate the proposed technique. Intensive practical results were obtained, which demonstrated the applicability of the proposed control approach to work in a robust manner with the ability of disturbance rejection. The practical results were presented in three different categories; angular speed and position control response on roll axis; angular speed and position control response on pitch axis; and angular speed and position control response on yaw axis. To prove the stable high performance balancing control systems on roll, pitch and yaw axes, different desired angular position command on each axis was applied.

A. Practical Results on Roll-Axis

This subsection shows the practical results for the balancing control system response on roll axis. It is presumed that there were no motion on the pitch axis i.e., the control balancing system on pitch axis tries to maintain $\theta=\dot{\theta}=0$ as well as no orientation around yaw axis i.e., $\psi=\dot{\psi}=0$.

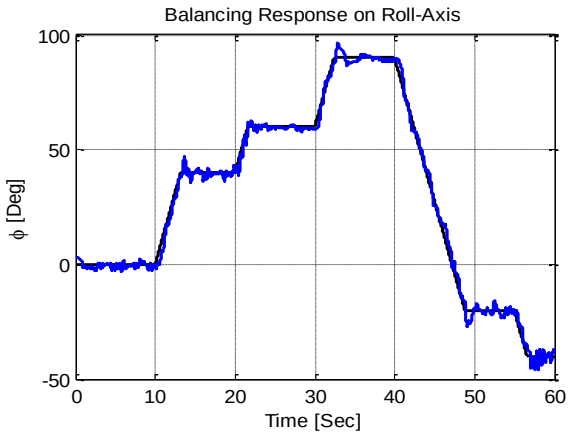


Fig. 8. Balancing Response on roll-axis

Fig. 8 shows the balancing system response on the roll axis. "Black" is the command signal, and "Blue" is the actual quadrocopter response. As seen the quadrocopter balancing system exhibited fast tracking response and zero steady state error. As noticed at time t=0➔t=10 sec, the system was balancing at zero roll angle.

B. Practical Results on Pitch-Axis

This section explains the dynamic performance of the quadrocopter balancing for pitch angle control.

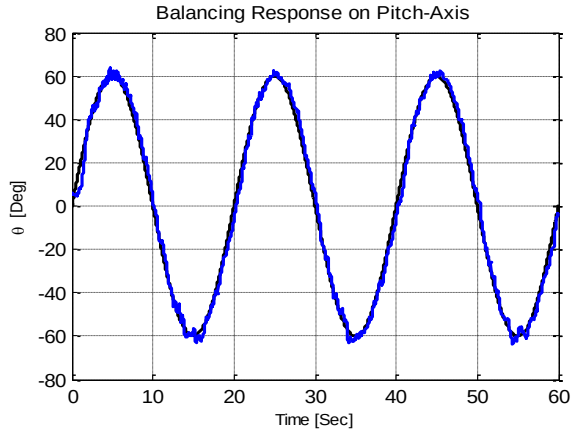


Fig. 9. Balancing Response on pitch-axis

Fig. 9 shows the response of the quadrocopter control for a sinusoidal changes in the desired command angle. As seen from Fig. 9, "black" is the command and "blue" is the actual system response. There is a very fast command tracking. It is worth mentioning that the peak variation of the quadrocopter around pitch axis is 60^o. Hence, the quadrocopter system exhibites a wavey motion with no lack of stability.

Fig. 10 shows the inner angular speed control response. As seen, the integral LQR conroller provides fast angular speed tracking for which the outer position control loop tracks fast the desired position command. Fig. 11 shows the control for the case when controlling the pitch angle, as seen by the angle command profile depicted in Fig. 9. The control law of Fig. 11 is to be sent to the four quadrocopter's rotors, for which stabilization and balancing as desired on pitch axis is achieved.

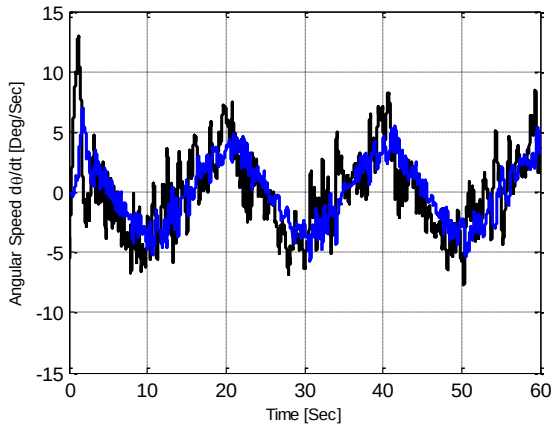


Fig. 10. Angular speed response on pitch-axis

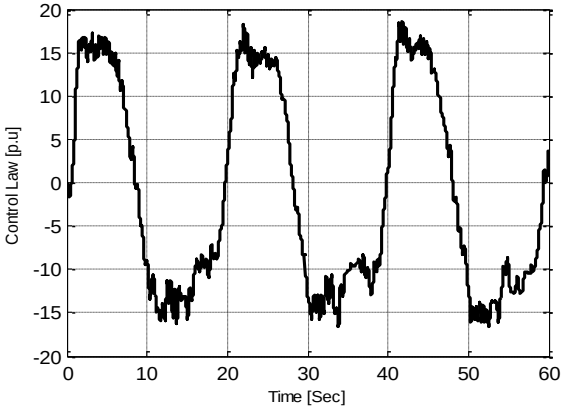


Fig. 11. Control law on pitch-axis

C. Practical Results on Yaw-Axis

The following section provides the experimental results of the quadrocopter balancing system response for yaw angle control.

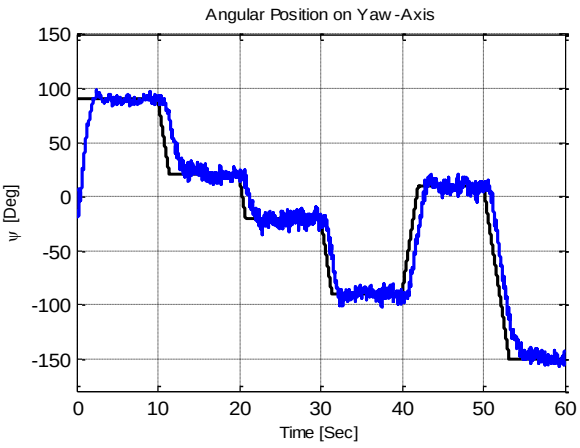


Fig. 12. Angular position response on yaw-axis

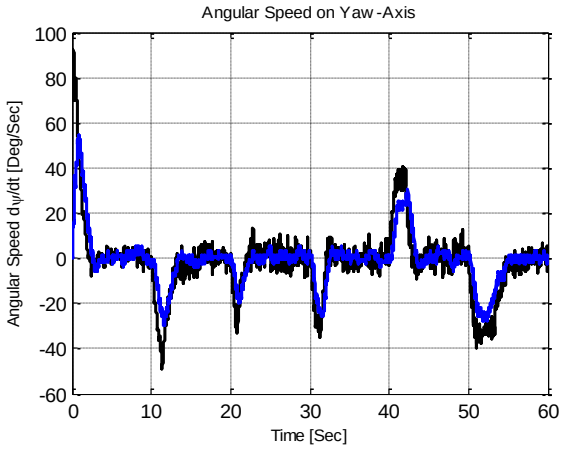


Fig. 13. Angular speed response on yaw-axis

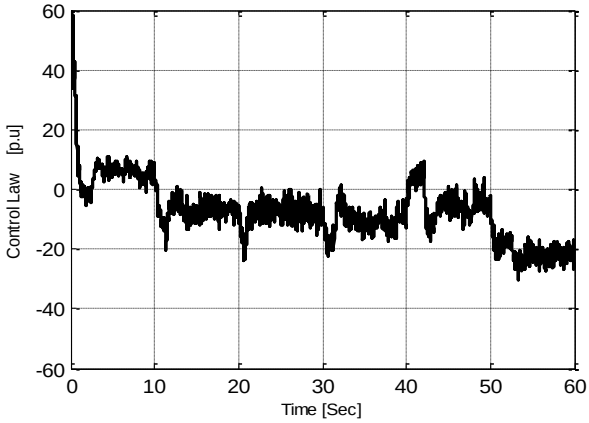


Fig. 14. Control law on yaw-axis

As seen, Fig. 12 and Fig. 13 show the system control response for the outer position control loop and the inner angular speed control loop respectively. “Black” is the command and “blue” is the actual system response. As shown, there is a robust control tracking in both the inner and the outer control loops. Fig. 14 shows the LQR control law generated to maitain stable operation and exact yaw angle tracking as it is desired.

Fig. 15. Real-Time Control of Balancing Quadrocopter System

Fig. 13 shows the test rig of the quadrocopter balancing control system. It has been completely developed by the graduate students in the integration of Mechatronics Systems Lab. As seen, the quadrocopter is linked with a four yellow straps. This is a safety procedure as the quadrocoper is running indoor. This procedure has no effect on the balancing system, as the system is due to work on the body frame angles control and has no motion with respect to the reference fixed earth frame.

VII. CONCLUSION

In this paper, it has been presented a practical implementation of a two-cascaded LQR control loops for position control of the quadrocopter's body frame angles. There are six control loops for the whole quadrocopter balancing control. An LQR and an integral LQR have been proposed for the outer loop (position loop). It has been shown that the proposed control approach, tracks fast the desired commands for roll, pitch and yaw angles in the body frame. It has been also noted that the proposed control approach, exhibits an inherited decoupling control action, for which the control of one axis angle has relieved the dynamic coupling effect on the other two axes. Furthermore, intensive practical results have demonstrated the robustness of the proposed controller. Future work is dedicated for the development of quadrocopter motion control on the Cartesian domain with respect to a fixed frame i.e., high-level control. Referencing to Fig. 4, a vision system assisted-novel SLAM technique is proposed to control and track the localization of the quadrocopter system. Real time implementation of the proposed SLAM technique is done utilizing an Intel® Core™ i5-480M computer machine with a wireless communication system. The communication protocol system is to provide a bi-directional paired communication between the quadrocopter on board control system and the SLAM approach hosted on the computer machine.

ACKNOWLEDGEMENT

The authors would like to thank the Syrian Society for Scientific Research (SSSR) for the financial support provided to cover the IJARAI registration fee

REFERENCES

[1] S. Bouabdullah "Design and Control of Quadrotors with Application to Autonomous Flying" EPFL, thesis No.3727, 129 pages, 2007.

[2] A. Eresen – N. İmamoğlu – M. Ö. Efe "Motion Detection and Tracking of classified Objects with Intelligent Systems" 2009 IFAC.

[3] W. Wang, H. Ma, C. -Y. Sun "Control System Design for Multi-Rotor MAV" Journal of Theoretical and Applied Mechanics, 51, 4, pp. 1027-1038, July 25-27 2013.

[4] A. A. Saif M. Dhaifullah M. Al-Malki M. El Shafie "Modified Backstepping Control of Quadrotor" 2012 9th International Conference on Systems, Signal and Devices.

[5] G. -V. Raffo M. -G. Ortega F. -R. Rubio "Backstepping/Nonlinear H∞ Control for Path Tracking of a QuadRotor Unmanned Aerial Vehicle" 2008 American Control Conference, 3356-3361 pp, June 11-13 2008.

[6] G. Cui – B. M. Chen – T. H. Lee " Unmanned Rotorcraft Systems" Springer, 267 pages, 2011.

[7] H. Khebbache M. Tadjine "Robust Fuzzy Backstepping Sliding Mode Control for a Quadrotor Unmanned Aerial Vehicle" CEAI, Vol.15, No.2, pp 3-11, 2013.

[8] T. Luukkonen, "Modelling and control of quadcopter," Independent research project in applied mathematics. pp. 2–4, School of Sience, Espoo, August 22, 2011.

[9] P. Castillo – R. Lozano – A. E. Dzul "Modeling and Control of Mini Flying Machines" Springer, 251 pages, 2005.

[10] R. Mahony, V. Kuar, P. Corke, "Multirotor Aerial Vehicles: Modeling, Estiation, and Control of Quadcopter," IEEE Robotics & Automation Magazine, vol. 19, no.3, pp. 20-32, Sept. 2012.

[11] A. C. Satici H. Poonawla M. W. Spong "Robust Optimal Control of Quadrotor UAVs" IEEE, Volume 1, pp 79-93, 2013.

[12] O. Santos, H. Romero, S. Salazar and R. Lozano, , "Discrete Optimal Control for a Quadrotor UAV: Experimental Approach," pp. 1139-1141, Orlando, FL, USA, May 27-30, 2014.

[13] K. Nonami - F. Kendoul - S. Suzuki - W. Wang - D. Nakazawa " Autonomous Flying Robots" Springer, 329 pages, 2013.

[14] L. R. G. Carillo - A. E. D. López - R. Lozano - C. Pégard " Quad Rotorcraft Control " Springer, 179 pages, 2013.

[15] H. Khalil " Nonlinear Systems" Prentice Hall, 680 pages, 2002.

[16] M. Krstić I. Kanellakopoulos P. Kokotović "Nonlinear Adaptive Control Design" Springer, 595 pages, 2010.

[17] C. Diao, B. Xian, X. Gu, B. Zhao, J. Guo "Nonlinear Control for an Underactuated Quadrotor Unmanned Aerial Vehicle with Parametric Uncertainties" proceeding of 31st Chinese Conference on Automatic Control, 998-1003 pp, July 25-27 2012.

[18] Z. Yaou Z. Wansheng L. Tiansheng L. Jingsong "The attitude control of the four-rotor unmanned helicopter based on feedback linearization control" WSEAS, Issue 4, Volume 12, April 2013.

[19] L. –X. Wang "Backstepping-Based Inverse Optimal Attitude Control of Quadrotor" Intech, Int J Adv Robotic Sy, 2013, Vol. 10, 223:2013.

[20] M. Önkol and M. Önder Efe. "Experimental Model Based Attitude Control Algorithms for a Quadrotor Unmanned Vehicle", TOBB Economics and Technology University, Turkey, pp 1-6.

[21] A. Ö. Kivrak, "Design of control systems for a quadrotor flight vehicle equipped with inertial sensors", master's thesis in Mechatronics Engineering Atılım University, December 2006, pp 1-37.

[22] M. Rich, "Model development, system identification, and control of a quadrotor helicopter",A thesis submitted to the graduate faculty in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE. Iowa State University Ames, Iowa 2012.

Cardiac Arrhythmia Classification by Wavelet Transform

Hadji Salah[1]
TIC department
LR SITI ENIT
BP 37 Belvédère 1002 Tunis, Tunisia

Ellouze Noureddine[2]
TIC department
LR SITI ENIT
BP 37 Belvedere 1002 Tunis, Tunisia

***Abstract*—Cardiovascular diseases are the major public health parameter; they are the leading causes of mortality in the world. In fact many studies have been implemented to reduce the risk, including promoting education, prevention, and monitoring of patients at risk. In this paper we propose to develop classification system heartbeats. This system is based mainly on Wavelet Transform to extract features and Kohonen self-organization map the arrhythmias are considered in this study : N,(Normal),V(PrematureVentricular),A(AtrialPremature),S(Extrasystolesupraventriculaire),F(FusionN+S),R(RightBundle Branch) .**

Keywords—ECG; QT database; Wavelet Transform; Classification; Kohonen self-organization map

I. INTRODUCTION

The ECG signal is a representation of the electrical heart activity; it is used to analyze the status of the heart. Due to the morphological variability of the different waves of ECG and the presence of noise that interfere with the ECG signal it is so difficult to extract necessary information's from the signal. These difficulties require tools of signal processing and automatic classification of cardiac Arrhythmia. Therefore, many researches were proposed for the automatic classification of the signal heart many methods have been implemented, such as the statistic approach, fuzzy logic and neural networks. In our paper, we have opted for neural classifiers because of their efficiencies were solving problems of classification in the case of large databases dimensions. As classifier, we chose to use the Kohonen topological maps [7,8,9] receives as input 12 parameters characterizing a temporal and morphological ECG beat are mainly: (Length (QRS), amp (Q, R,S) intervals (QT, PR, RR)). In this article, we present in Section II the steps and algorithms used for extracting the characteristic parameters of a heart beat. In Section III, we will describe the methodology adopted for the development of our system classification. In Section IV, we present the main results. A conclusion and outlook for this work are given in Section V.

II. WAVELET TRANSFORM

Wavelet Transform is a time-scale representation of signals that has been used in a wide range of applications, including signal analysis and compression. Recently, wavelets have been applied to several problems in Electro cardiology, including data compression, analysis of ventricular late potentials and time localization of ECG characteristics. The Wavelet Transformation (WT) is a linear operation that decomposes a signal into a number of scales related to frequency components and represents each scale with a fixed time and frequency resolution [12]. WT analysis uses a short time interval for evaluating high frequencies and a long time interval for low frequencies. Wavelet transform of a signal f(t) is defined as the sum of over all time of the signal multiplied by scaled versions of the wavelet function ψ, and is given by :

$$w_{a,b} = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t)\delta t \quad (1)$$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi^{*}\left(\frac{t-b}{a}\right) \quad (2)$$

Where * denotes complex conjugation of the mother wavelet $\psi_{a,b}(t)$ and $\psi\left(\frac{t-b}{a}\right)$ is the shifted and scaled version, the constant $\frac{1}{\sqrt{a}}$ is an energy normalization factor. When the scale parameter is the set of integral powers of 2, i.e. $a=2^{j}$ ($j \in z$, z is an Integer set). The wavelet transform is called dyadic wavelet transform. The wavelet transform at scale 2^{j} is given by:

$$Wf(2^{j},\tau) = \frac{1}{\sqrt{2}}\int_{-\infty}^{+\infty} f(t)\left(\frac{t-\tau}{2^{j}}\right)\delta\tau \quad (3)$$

A. Stationary wavelet transform

The stationary wavelet transform (SWT: Stationary Wavelet Transform) is an intermediate representation of views between the redundant CWT high redundancy and non-redundant DWT. It retains an almost dyadic sampling continuous and uniform time [12] [59], the coefficients of its decomposition have the appearance of the filtered signals in contrast to the DWT [12]. The SWT is defined by calculating the coefficients: $cd_{x}(j,k)_{(j,k)\in z^{2}}$

$$cd_{,x}(j,k) = \int_{R} x(t)\psi^{*}{}_{j,k}(t)dt \quad (4)$$

Where $\psi(t) = 2^{\frac{-j}{2}}\psi_{0,0}\left(2^{-j}(t-k)\right)$
the relationship between the wavelet bases used in the DWT and SWT is implemented using the following wording:

$$\psi_{j,k}\psi_{j,0}\left(t-2^{j}k\right), \psi'_{j,k}(t) = \psi_{j,0}(t-k) \quad (5) \text{ with}$$

$$\psi_{j,o}(t) = 2^{\frac{-j}{2}} \psi\left(\frac{t}{2^j}\right)$$

and thus the SWT coincides with the CWT at the uniform grid and the grid dyadic DWT where $cd_x(j,k)$ have the same frequency resolution as dx (j, k), but sharing information on the time axis

$$cd_x(j,k) = cwt_x(a = 2^j, t = k) \qquad (6)$$
$$cd_x(j,2^j k) = d_x(j,k)$$

The SWT is. Translation invariant in time, what is missing in the DWT and these results from the fact that it has a time sampling of the same signal. This explains the cd_j have the properties of the filtered signal

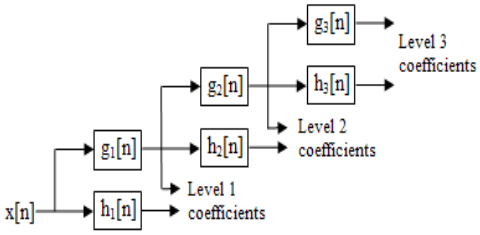


Fig. 1. 3 level decomposition

B. QT database:

The QT is a basic standard database available on the website PhysioNet, which is an international reference in the field of ECG signal processing [11]. The database QT PhysioNet was established by researchers to be used as a reference for validation and comparison of algorithms for segmentation of the ECG. It is known to have a low signal-to-noise ratio with different pathologies leading to accurate detection of waves through this entire base. Therefore, at present, the basic QT is the only database that is annotated appropriately to test our analysis method of long-term recordings. It contains 105 two-channel Holter-recordings primarily (ITNs and V5) of 15 minutes sampled at 250 Hz. These ECG signals were extracted from different databases already existing such as the database arrhythmias MIT-BIH, the database ST-T of the European Society of Cardiology, and several other databases assembled by Boston's Beth medical center

III. SOM SELF-ORGANIZATION MAP

SOM (Self-organizing Map) is an artificial neural network (ANNA) architecture based on unsupervised, competitive learning [14]. It provides a topology preserving, smooth mapping from a high-dimensional input space to the map units usually arranged as a two-dimensional lattice of neurons (nodes). Thus, the SOM can serve as a tool for cluster analysis of complex, high-dimensional data. A parametric reference vector m is associated with every node. A data vector x is compared to all reference vectors in any metric and the best matching node is defined, e.g., by the smallest Euclidean distance between the data vector and any of the reference vectors. During learning, those nodes that are topographically close in the array up to a certain distance will activate each other to learn from the same input:

$$m_i(t+1) = m_i(t) + h_{ci}(t).[x(t) - m_i(y)] \qquad (7)$$

where t is an integer representing time, and h_{ci} is the so-called neighborhood kernel describing the neighborhood that is updated around the best-matching node. Several suitable kernels can be used, e.g. a so-called bubble kernel or a Gaussian kernel, relating to different ways of determining the activating cells. The kernel also includes the learning rate parameter) (t α. With time, the size of the neighborhood and the learning rate are diminished. The described learning process leads to a smoothing effect on the weight vectors in the neighborhood and by continued learning to global ordering of the nodes [14]

The SOM consists of a two-dimensional lattice that contains a number of neurons. These neurons are usually arranged in a rectangular or hexagonal way. The position of the units in the grid, especially the distances between them and the neighborhood relations, are very important for the learning algorithm. A prototype vector (also "model" or "codebook" vector) is associated with each neuron, which is a vector of the same dimension as the input data set. This prototype vector approximates a subset of the sample vectors. The dimension of the sample is called input dimension, and is usually larger than 2, the dimension of the lattice, which is called output dimension. For training and visualization purposes, the sample vectors are assigned to the most similar prototype vector, or best-matching unit (BMU), formally

$$C(x) = \arg\min\{|x - m_i(t)|\} \qquad (7)$$

The learning process itself gradually adapts the model vectors to match the samples and to reflect their internal properties as faithfully as possible, which means that input vectors which are relatively close in input space should be mapped to units that are relatively close on the lattice.

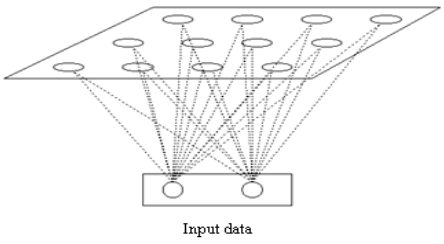


Fig. 2. Kohonen Network map

A. Parameters extraction

Each ECG beat, consisting of 376 samples centered on the position of the R wave undergoes a SWT on four successive scales using wavelet (db4) as mother wavelet. Subsequently, we calculate the energy of the Stationary wavelet transform each level. Thus it reduces the size of the input vector of 376 elements only 8 for the 4th level gave the best recognition rate. The classification scheme is shown in Figure 3.

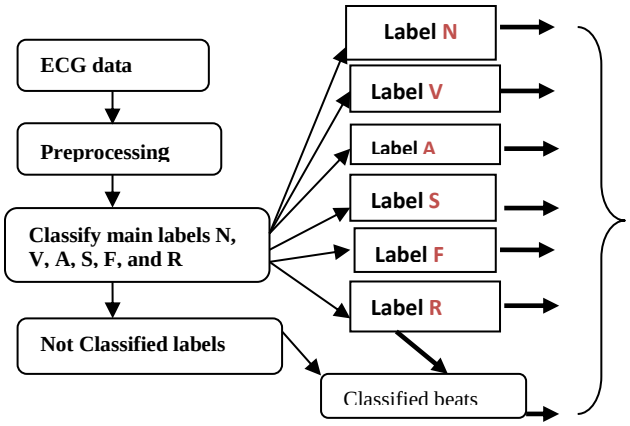


Fig. 3. Classification schème

B. *Performance and Measures*

The performance of the classification system was measured based on tow standard statistical measures for each class: Sensitivity (Se) and Predectivity (Sp) which are calculated from multi-class classification table [1].

IV. RESULTS

TABLE I. CONFUSION- MATRIX

	N	V	A	S	F	R
N	26792	23	455	458	125	360
V	0	562	0	0	35	6
A	0	0	365	0	4	3
S	1	5	1	235	3	1
F	2	0	0	0	58	0
R	4	3	0	0	0	123

	Se (%)	**Sp** (%)
N	94,7	99,98
V	93,7	94,7
A	97,4	98,3
S	95,4	98,3
F	96,6	96.6
R	94,4	94,1

Fig. 4. displayed results with SOM

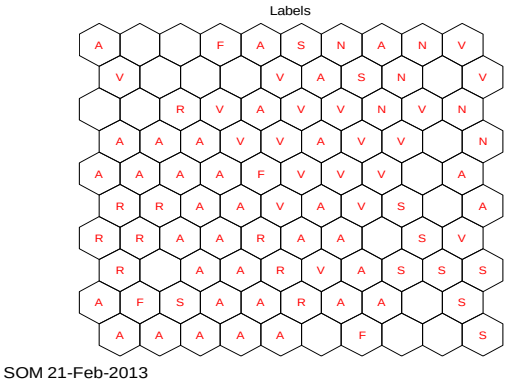


Fig. 5. cardiac Arrhythmia recognition and displayed by Kohonen map

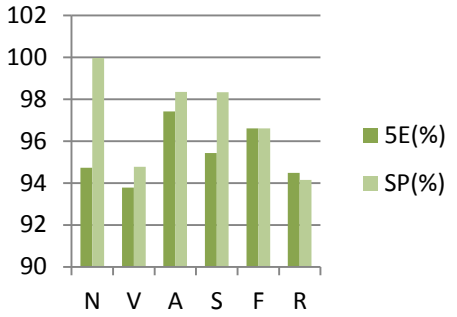


Fig. 6. Histogram variation of recognition rates by wavelet Transform

V. CONCLUSION

In this work we have developed a new method based on wavelet transform decomposition and self organizing map algorithm to classify cardiac arrhythmia labels such as N, A, V, R, F, S, obtained results are very interesting .

The specificity and the Predectivity were higher than 94% in all cases (N, A, S, R, V and F) .This result means that the proposed method is suitable enough for practical clinical use.

REFERENCES

[1] Physiobank Archive Index, QT Database. http://www.physionet.org/physiobank/database

[2] S.Z.Mahmoodabadi,A.Ahmadian and M D. Abolhasan,"ECG Feature Extraction using Daubechies Wavelets", Proceedings of the fifth IASTED International conference on Visualization, Imaging and Image Processing, pp. 343-348, 2005.

[3] Palreddy, S., Hu, Y.H. and Tompkins, W.J. "A patient adaptable ECG beat classifier using a mixture of experts approach", IEEE Trans. on BME, 44(9), 1997, 891–900.

[4] Acr, N. "Classification of ECG beats by using a fast least square support vector machines with a dynamic programming feature selection algorithm", Neural Computing and Applications, 14(4), 2005, 299–309.

[5] P. de Chazal, M. O'Dwyer, and R. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," IEEE Trans. Biomed. Eng., vol. 51, no. 7, pp. 1196–1206, Jul. 2004.

[6] Übeyli, E.D., ECG beats classification using multiclass support vector machines with error correcting output codes Digital Signal Processing, Volume 17, Issue 3, May 2007, 675-68

[7] Osowski, S. and Linh, T.R. "ECG Beat Recognation Using Fuzzy Hybrid Neural Network", IEEE Trans. on BME, 48(11), 2001.

[8] Nadal, J. and Bossan, M. "Classification of cardiac arrhythmia based on principal components analysis and feedforward neural Networks", Comput Cardiol, 1993, 341–344.

[9] Hosseini, H.G., Reynolds, K.J., and Powers, D., "A Multi-stage Neural Network Classifier for ECG Events", Proceedings of the 23rd International Conference of the IEEE Engineering in Medicine and Biology Society, 2001, October 25-28.

[10] Minami, K., Nakajima, H. and Toyoshima, T. "Arrhythmia diagnosis with discrimination of rhythm origin and measurement of heart-rate variation", Comp Cardiol,1997,243–246.

[11] Kohonen, T., (2001). Self-Organizing Maps, 3rd ed., Heidelverg: Springer-Verlag Berlin.

[12] Salah Hadji, "Caractérisation du complexe QRS du signal ECG et Identification des arythmies cardiaques'', Thèse de doctorat de l'université de Tunis Elmanar 3,2012

[13] Kohonen, T.: Self-organizing Maps, Springer-Verlag,1995.

[14] [1Kohonen,T.,Hynninen,J.,Kangas,J.Laaksonen,J.:SOM_PAK - The Self-organizing Map Program Package, 1995.

A Novel Control-Navigation System-Based Adaptive Optimal Controller & EKF Localization of DDMR

Dalia Kass Hanna, Bsc, Msc(Candidate)
Mechatronics Department
University of Aleppo
Aleppo-Syria

Abdulkader Joukhadar, Bsc, MPhil, PhD
Mechatronics Department
University of Aleppo
Aleppo-Syria

***Abstract*—This paper presents a newly developed approach for Differential Drive Mobile Robot (DDMR). The main goal is to provide a high dynamic system response in the joint space level, the low level control, as well as to enhance the DDMR localization. The proposed approach depends on a Linear Quadratic Regulator (LQR) for the low level control and an Adaptive LQR for the high level control. The investigated DDMR is considered highly nonlinear system due to uncertainty exhibited by the mobile robot incorporated with actuators nonlinearity. DDMR's uncertainty leads to erroneous localization. An Extended Kalman Filter (EKF) -based approach with fusion sensors is used to enhance the robot degree of belief for its posture. Intensive simulation results obtained from the developed uncertain model and the proposed approach have shown very good dynamic performance on the low level control and very good convergence to the desired posture of the mobile robot path with the presence of robot uncertainty.**

Keywords—DDMR modelling; Localization; LQR; Adaptive LQR; EKF; System Uncertainty

I. INTRODUCTION

The question of "where am I?" exhibited by mobile robots, in general, remains challenging and incompletely covered in academics. The topic of mobile robot localization has been paid wide attention by academics and industry to enhance the robot performance in different aspects for which the robot can perform its motion towards any desired posture with as minimum error as possible. Different techniques proposed to enhance trajectory tracking of wheeled mobile robots. [1, 2, 3, 4] utilize EKF algorithm with fusion sensors and gyroscope to localize the robot system as close as possible to the desired posture. The disadvantages of the work presented in [1, 2] do not pay attention to the dynamic performance of the system due to robot uncertainty but they considered only the noise affecting the proprioceptive and the exteroceptive sensors.

However, in [2, 3] researchers use fusion sensors and vision sensor to help the system to be located correctly in a specific location. People in [3] implement EKF-based algorithm assisted with a gyroscope sensor to enhance the robot localization. Researchers in [3, 4] consider the proprioceptive sensors provide exact information about the robot motion in which the robot's posture is considered correct but noisy. Thus [3, 4] use EKF to purify the information about the robot's localization. [6, 8] have proposed trajectory tracking algorithm but considered the information comes from the proprioceptive sensors is correct enough to determine the robot's posture with no lack of accuracy. The uncertainty of the mobile robot due to inaccuracy in the mechanical robot design and due to joint space inaccuracy exhibited by the mobile robot actuators lead to accumulated drift and divergence from the desired robot's posture.

This paper focuses on a novel control approach utilizing Linear Quadratic Regulator for joint space control of robot actuators as well as proposing EKF-assisted optimal controller to overcome the problem of robot uncertainty, which may lead to robot posture divergence. The proposed approach is supported by fusion sensors consisted of a gyroscope sensor (rate & accelerometer) which is fixed to the robot's center of gravity and robot's on board sensors (odometry sensors). The onboard sensors (proprioceptive sensors) assumed noisy, in addition to the robot's mechanical parameters also considered highly uncertain.

It is presumed, as well, that the mobile robot is due to some random disturbance represented by τ_d which is very common in the system control areas [3]. The remaining sections of the paper are as follows: Section II provides details about the dynamic model of the mobile robot incorporating the actuators dynamics; section III explains the design of the proposed controller for the joint space control system i.e., low level control; section IV discusses mobile robot navigation and localization; section V exhibits intensive simulation results with different system uncertainty and section VI provides a conclusion for the presented work.

II. DIFFERENTIAL DRIVE MOBILE ROBOT MODEL

A. Mobile Robot Motion Description:

The HBE-RoboCar wheeled mobile Robot (WMR) has four driving wheels, which determine the moving direction of the robot through the rotating direction and speed of wheels. This WMR uses 4-DC geared motors for operation, and each wheel has one motor mounted. The wheels in the same side (left or right) are operated together as shown in Fig.1 to follow a certain robot trajectory. Such wheeled robots called Differential Drive Mobile robot (DDMR). In this section a mathematical description of DDMR moving on a planar surface is presented.

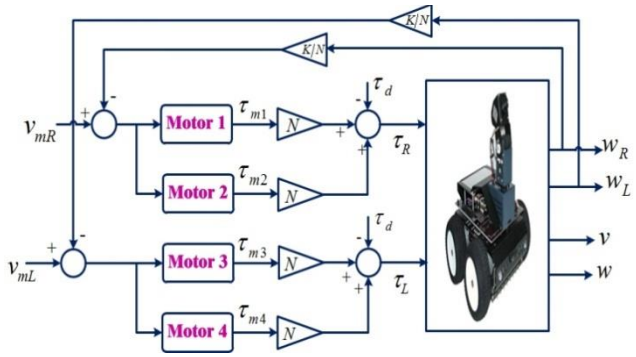


Fig.1. Block diagram of DDMR with actuators.

Usually, the DDMR's posture is determined in its environment based on two coordinate frames: the Global frame {G} and Local Frame {L}. The global frame is fixed in the environment in which the robot moves in. The local frame attached to the DDMR at the middle point A between two back wheels. The movement of point A represents the movement of the robot [9].

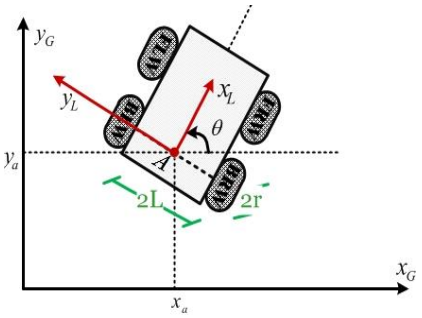


Fig.2. DDMR Coordinates Systems

As shown in Fig.2, x_a and y_a coordinate denote the position of DDMR in the global frame. The angle θ between the moving direction of the DDMR and the positive direction of the x-axis of the global coordinate frame denotes the orientation. Symbols used in this section are listed in Table1:

TABLE I. KINEMATIC & DYNAMIC MODEL VARIABLES:

Parameter:	***Description:***
$2L$	Distance between two wheels (m)
C	The center of mass
A	the middle point between two back wheels
d	Distance between A and C (m)
r	Wheel radius (m)
ω_R, ω_L	The angular velocity of the wheels ($rad.s^{-1}$)
v	The linear velocity of DDMR ($m.s^{-1}$)
ω	The angular velocity of DDMR ($rad.s^{-1}$)
N	Gear ratio
K	back *emf* constant ($vs.rad^{-1}$)
M	Robot mass (kg)
J	Robot moment of inertia ($kg.m^2$)
F_{uR}, F_{uL}	Tangential forces exerted on DDMR by the wheels.
F_{wR}, F_{wL}	Radial forces exerted on DDMR by the wheels.

The study of DDMR motion is divided into three parts including kinematics, dynamics and drivers [8, 9].

B. Kinematic Model of the DDMR:

Kinematics is the most basic study of how mechanical systems behave in order to model, analyze and simulate any control system design. The motion of this type of wheeled robot is classified as non-holonomic, which means, the motion constraint equations are needed to introduce into the DDMR's motion equations based on two main assumptions [9]:

- The robot can't move sideward. These non-holonomic constraints are taken into account by defining the velocity of the center point A in the local frame and forcing it to be zero. This constraint introduces in the global frame related to the velocity of point A by the following equation (1):

$$-\dot{x}_a^G \sin(\theta) + \dot{y}_a^G \cos(\theta) = 0 \quad (1)$$

Where $q^G = \begin{bmatrix} x_a^G & y_a^G & \theta \end{bmatrix}^T$ is the coordinate of point A in the global frame.

- To simplify the model, it is assumed that each wheel has one contact point with the ground and there is no slippage in its longitudinal axis and lateral axis. The velocities of the contact points in the local frame are related to the wheel velocities by (2):

$$\begin{aligned} V_R &= r\omega_R \\ V_L &= r\omega_L \end{aligned} \quad (2)$$

The linear and angular velocities of the DDMR related to the wheels velocities and the geometric parameters of the robot are given as follows (3):

$$\begin{aligned} v &= (V_R + V_L)/2 \\ \omega &= (V_R - V_L)/2L \end{aligned} \quad (3)$$

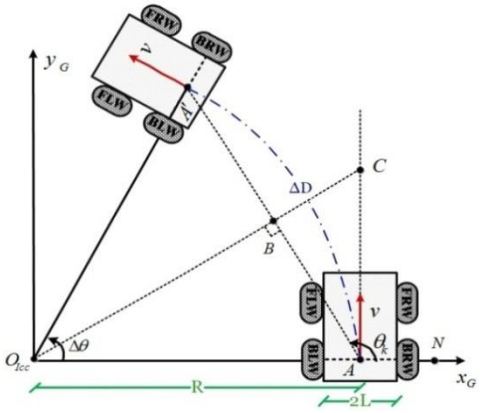


Fig.3. The DDMR new position and orientation

To determine the new DDMR's position and orientation it is considered that the robot moves from the point A with a position (x_{-1}, y_{-1}), and orientation ($\theta_{-1} = \angle NAC$) to A' through a circular arc with an Instantaneous Center of Curvature (ICC) [2].

The increment of distance denoted by (ΔD) and orientation denoted by ($\Delta\theta$). As shown in Fig.3, based on triangular relationship the angle ($\angle CAB = \Delta\theta/2$) and the new orientation of DDMR given by (4):

$$\theta = \theta_{-1} + \Delta\theta/2 \tag{4}$$

Considering that the increment of distance and orientation is small then $AA' = \Delta D$ and the DDMR kinematic equation is given by (5):

$$q_{=}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} x_{-1} + \Delta x \\ y_{-1} + \Delta y \\ \theta_{-1} + \Delta\theta \end{bmatrix} = \begin{bmatrix} x_{-1} + \Delta D\cos(\theta_{-1} + \Delta\theta/2) \\ y_{-1} + \Delta D\sin(\theta_{-1} + \Delta\theta/2) \\ \theta_{-1} + \Delta\theta/2 \end{bmatrix} \tag{5}$$

Equation (5) is called Navigation Equation also. The navigation system analysis and design are based on equation (5). Another form for the kinematic model describes the robot behavior relative to the linear and angular velocities of the DDMR which can be written as follows (6):

$$\begin{bmatrix} \dot{x}_A^G \\ \dot{y}_A^G \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{6}$$

The velocities of the center of mass C represented in the global inertial frame are given by (7):

$$\begin{bmatrix} \dot{x}_C^G \\ \dot{y}_C^G \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{7}$$

Equation (7) is the relation between the velocities in local and global frames named as Guidance matrix G .

C. Dynamic Model of the DDMR:

In this subsection, the dynamic behavior of DDMR mechanisms based on Newton-Euler approach is presented [9].

By considering the free body diagram of DDMR; Fig.4 shows the forces acting on the DDMR [9].

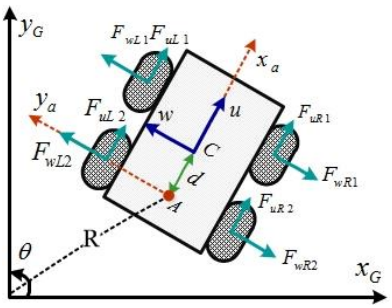


Fig.4. Active Forces of the DDMR.

From Fig.4 the DDMR position, velocity and acceleration represented in the global frame using polar coordinate system and the equations (8) are divided using Newton's laws of motion in the robot frame:

$$\begin{aligned} M\dot{v} - Md\omega^2 &= 2(\tau_R + \tau_L)/r \\ (Md^2 + J)\dot{\omega} + Md v\omega &= 2L(\tau_R - \tau_L)/r \end{aligned} \tag{8}$$

where F_{uR}, F_{uL} is linearly dependent on the wheel control input as follows:

$$F_{uR} = \tau_R/r, F_{uL} = \tau_L/r$$

Equations (8) represent the dynamic model of DDMR taking into account the non- holonomic constraints [9].

III. Controller System Design:

In this section, a controller design based on solving quadratic optimal control problem has been presented to improve the dynamic performance of the DDMR for accurate trajectory tracking.

Classical control methods are first designed and then their stability is examined. In optimal control, based on Liapunov approach the conditions for stability are formulated first and then the system is designed within these limitations. Thus, the designed system has a configuration with inherent stability characteristics [12].

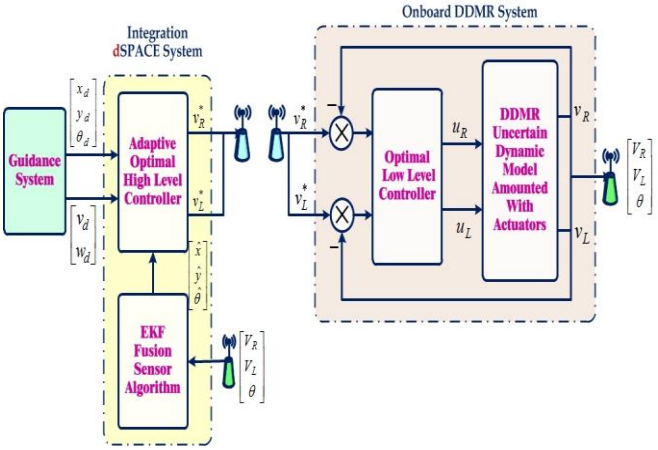


Fig.5. Optimal High and Low Level Control Structure Based DDMR.

The proposed control scheme shown in Fig.5 has two main controllers; high level control based on kinematic model of DDMR, that; correct the robot position and orientation to follow the commanded trajectory; and low level control based on dynamic model of DDMR which follows the velocities commands given by the high level controller. Both controllers employ quadratic optimal control approach.

A. Low Level Controller:

It is presumed in subsection (II, B) that, the linear and angular velocity of the DDMR related to the wheels velocities are given by (3) then the accelerations terms of the robot velocity and orientation angular speed are given by (9):

$$\begin{aligned} \dot{v} &= (\dot{V}_R + \dot{V}_L)/2 \\ \dot{w} &= (\dot{V}_R - \dot{V}_L)/2L \end{aligned} \tag{9}$$

By substituting (9) equations in the dynamic equation of DDMR (8):

$$\dot{V}_R = 2A\tau_R + 2B\tau_L - \eta_1 + \eta_2$$
$$\dot{V}_L = 2B\tau_R + 2A\tau_L + \eta_1 + \eta_2 \quad (10)$$

Where η_1, η_2 are the coupling terms between the left and the right wheels. An integral optimal control is proposed to control right and left wheels velocities by inserting an integrator in the feed forward path between the error comparator and the plant.

Fig.6 shows the block diagram of the DDMR with optimal low level control.

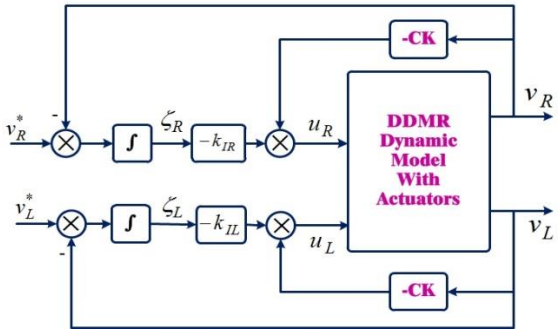


Fig.6. DDMR with optimal low level control.

The vector control u_{op} which minimizes a selected cost function J given in (11) is determined as follow to solve the quadratic optimal control problem for the system given in (9):

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (11)$$

Where:

$x_{4\times1}$: System state vector.

$u_{2\times1}$: Vector control.

$Q_{4\times4}$: Positive semi-definite symmetric matrix determines the relative importance of the error.

$R_{2\times2}$: Positive definite symmetric matrix determines the relative importance of the expenditure of the energy of the control signals.

The optimal control law is given as follows (12):

$$u_{op} = -R^{-1} B_1{}^T P x \quad (12)$$

$P_{4\times4}$ is the state covariance matrix which can be obtained from Reccati equation which is given as follows (13):

$$A^T P + PA - PB_1 R^{-1} B_1{}^T P + Q = 0 \quad (13)$$

By designing the low level control system to have exponentially stable dynamic response with high dynamic performance response, the high level controller can be designed, considering linear dynamic response in the low level.

B. High Level Controller:

A proposed Adaptive Discrete Quadratic Optimal Controller formula based on DDMR kinematic system is described in this subsection. To derive the high level controller, the mathematical model of DDMR kinematic will be used. As seen the velocities of the center of mass C represented in the global frame are given by (7).

It should be noted that the kinematic model of this type of wheeled robot is classified as nonlinear and involves non-holonomic constraints. To apply the optimal controller, a linear model of DDMR kinematic needs to be obtained. But if the linearization was about a stationary operating point the system become not controllable (In case that $\theta = 0$ and the DDMR move straightforward the information about y axes is lost. A novel technique is developed and applied on DDMR) by Adapting the discrete quadratic optimal algorithm to include this type of nonlinear systems using linear error model system along a desired trajectory [7].

First, equation (7) is discretized using the backward Euler method with sampling time T_s. The discretized form of DDMR kinematic model is obtained as shown in (14):

$$\begin{aligned} x_k &= (v_k \cos\theta_{k-1} - dw_k \sin\theta_{k-1})T_s + x_{k-1} \\ y_k &= (v_k \sin\theta_{k-1} + dw_k \cos\theta_{k-1})T_s + y_{k-1} \\ \theta_k &= w_k T_s + \theta_{k-1} \\ or &: \\ x_k &= f(x_{k-1}, u_k) \end{aligned} \quad (14)$$

where $x_k = [x_k \quad y_k \quad \theta_k]^T$ is the system state vector and $u_k = [v_k \quad w_k]^T$ is the input vector.

Second, defining the same equations for a desired trajectory generated by Guidance System as follows (15):

$$x_{dk} = f(x_{dk-1}, u_{dk-1}) \quad (15)$$

The posture error vector e_{xk} is given by (16):

$$[e_{xk} \quad e_{yk} \quad e_{\theta k}]^T = [x_{dk} \quad y_{dk} \quad \theta_{dk}]^T - [x_k \quad y_k \quad \theta_k]^T \quad (16)$$

The velocities error vector e_{uk} is given by (17):

$$[e_{vk} \quad e_{wk}]^T = [v_{dk} \quad w_{dk}]^T - [v_k \quad w_k]^T \quad (17)$$

The linear error model is given by (18):

$$e_{xk} = F_x e_{xk-1} + F_u e_{uk} \quad (18)$$

Where:

$$F_x = \begin{bmatrix} 1 & 0 & (-v_{dk}\sin(\theta_{dk}) - dw_{dk}\cos(\theta_{dk}))T_s \\ 0 & 1 & (v_{dk}\cos(\theta_{dk}) - dw_{dk}\sin(\theta_{dk}))T_s \\ 0 & 0 & 1 \end{bmatrix}$$
$$F_u = \begin{bmatrix} T_s\cos(\theta_{dk}) & -dT_s\sin(\theta_{dk}) \\ T_s\sin(\theta_{dk}) & dT_s\cos(\theta_{dk}) \\ 0 & T_s \end{bmatrix} \quad (19)$$

F_x, F_u are the Jacobian matrices of the system with respect to e_{xk-1}, e_{uk} respectively which was derived around a desired trajectory. Now, considering the following discrete cost function:

$$J = \sum_0^\infty e_k^T Q e_k + u_k^T R u_k \quad (20)$$

The closed loop system state space will be (21):

$$e_{xk} = (F_x - F_u K) e_{xk-1} \quad (21)$$

Where K is the optimal control gain vector. These errors in position and orientation of DDMR are represented in the global frame, then the errors in the local frame have been found as follows (22):

$$e_{uk} = G^{-1} e_{xk} \quad (22)$$

The velocities commands which tracked by the low level control given by (23):

$$\begin{aligned} v_R^* &= (v_{dk} + e_{vk}) + L(w_{dk} + e_{wk}) \\ v_L^* &= (v_{dk} + e_{vk}) - L(w_{dk} + e_{wk}) \end{aligned} \quad (23)$$

IV. Navigation System and Localization:

The robot navigation is the task of an autonomous robot to move safely from one location to another [13]. To make a truly autonomous robot an accurate localization is a key problem for successful navigation systems [4, 11].

The objective is to accurately determine DDMR's posture involving sensor noise uncertainties and potential failures in an optimal way with respect to a global or local frame of reference by integrating (fusing) kinetic information received from proprioceptive sensors (odometry and gyroscope) which give the robot feedback about its driving actions and obtaining knowledge about DDMR's environment [13].

In mobile robot navigation systems, onboard navigation sensors based on dead-reckoning are widely used. Dead reckoning is the process of calculating DDMR's current position by using a previously determined position and estimated speeds over the elapsed time [2, 5].

In the present section, EKF sensor fusion method is used for the estimation of DDMR's accurate posture and eliminates the effect of uncertainty associated with the system.

The Extended Kalman filter (EKF) is a recursive optimum stochastic state estimator which can be used for parameter estimation of a non-linear dynamic system in real time by using noisy monitored signals that are disturbed by random noise [14, 15, 16].

The EKF has been widely used for mobile robot navigation and system integration to address the nonlinearity in the system kinematic.

The goal of the EKF is to estimate the unmeasurable state (e.g. DDMR's posture) by using measured states, and also statistics of the noise and measurement (i.e. covariance matrices Q, R, P of the system noise vector, measurement noise vector, and system noise vector respectively) [15, 16, 17]. So the problem of mobile robot localization can usually not be sensed directly. The robot has to integrate data over time to determine its pose as accurate as possible using EKF fusion sensors method [4].

In order to apply the EKF algorithm [15], first, a discrete time Navigation Model for the DDMR based on equation (6) using the backward Euler method with sampling time T_s has been obtained (24):

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = (1/2) \begin{bmatrix} T_s\cos(\theta) & T_s\cos(\theta) \\ T_s\sin(\theta) & T_s\sin(\theta) \\ T_s/L & -T_s/L \end{bmatrix} \begin{bmatrix} V_{Rk} \\ V_{Lk} \end{bmatrix} + \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} \quad (24)$$

Second, note that equation (24) is nonlinear. So, in order to apply EKF algorithm, a linear approximation of Navigation Equation using Taylor series have to be obtained as follows (25):

$$x_k = J_x x_{k-1} + J_u u_k \quad (25)$$

where:

$$x_k = [x_k \quad y_k \quad \theta_k]$$
$$u_k = [V_{Rk} \quad V_{Lk}]$$

The terms J_x, J_u are the Jacobian which are obtained by differentiating equation (24) with respect to the state vector and input vector respectively (26):

$$J_x = \begin{bmatrix} 1 & 0 & -v_k T_s \sin(\theta) \\ 0 & 1 & v_k T_s \cos(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$
$$J_u = (1/2) \begin{bmatrix} T_s(\cos(\theta) - \frac{v_k}{2L}\sin(\theta)) & T_s(\cos(\theta) + \frac{v_k}{2L}\sin(\theta)) \\ T_s(\sin(\theta) + \frac{v_k}{2L}\cos(\theta)) & T_s((\sin(\theta) - \frac{v_k}{2L}\cos(\theta)) \\ T_s/L & -T_s/L \end{bmatrix} \quad (26)$$

The EKF algorithm contains basically two main stages [10], Fig.7:

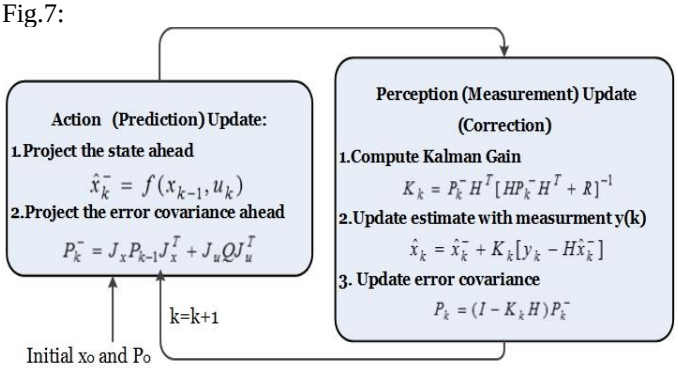


Fig.7. EKF Fusion Sensors Recursive Algorithm.

1) Action (or prediction) update:

Having a priori knowledge implies that x_{k-1}, P_{k-1} are initializing, the robot moves and estimates its position through its ideal proprioceptive sensors u_k.

$$\hat{x}_k^- = f(x_{k-1}, u_k) \quad (27)$$

The covariance matrix after moving is given by (28):

$$P_k^- = J_x P_{k-1} J_x^T + J_u Q J_u^T \quad (28)$$

During this step, the robot uncertainty grows.

2) Perception (or measurements) update:

The robot makes an observation using its uncertain model and corrects its position by opportunely combining its belief before the observation with the probability of making exactly

that observation. The Kalman gain is chosen to minimize the estimation error variance of the states to be estimated.

$$K_k = P_k^- H^T [HP_k^- H^T + R]^{-1} \quad (29)$$

The predicted state estimate $\hat{x}_k$ (and also its covariance matrix P_k) is corrected recursively through a feedback correction scheme which is the product of the Kalman gain K and the deviation of the estimated measurement output vector and the actual output vector ($y - \hat{y}$) that makes use of the actual measured quantities.

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k [\, y_k - H\hat{x}_k^- \,] \\ P_k &= (I - K_k H) P_k^- \end{aligned} \quad (30)$$

During this step, the DDMR uncertainty shrinks. If one looks at the problem of probabilistically, one can say that the robot has a degree of belief (*bel*) about where it is [5].

The goal of localization is to make this belief get as close as possible to the real distribution of the robot location. The robot incorporates these measurements into its belief to form a new belief about where it is [5].

The determinant of P_k provides a good measure of uncertainty as it is proportional to the volume of the deviation error ellipsoid. [5]

The degree of belief is given by (31):

$$bel = 1 - \sqrt{\det(P_k)} \quad where \quad 0 < bel \leq 1 \quad (31)$$

The higher $det(P_k)$ is the less degree of belief there is in the measurements.

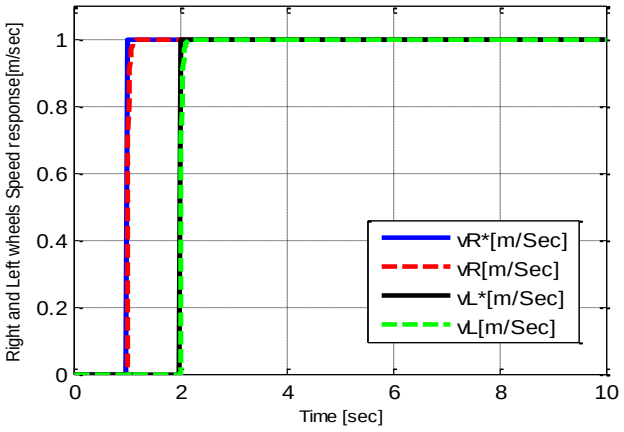
V. Simulation Results:

This section, several test results are demonstrated for the proposed control-navigation system based DDMR using MATLAB/ SIMULINK and C code.

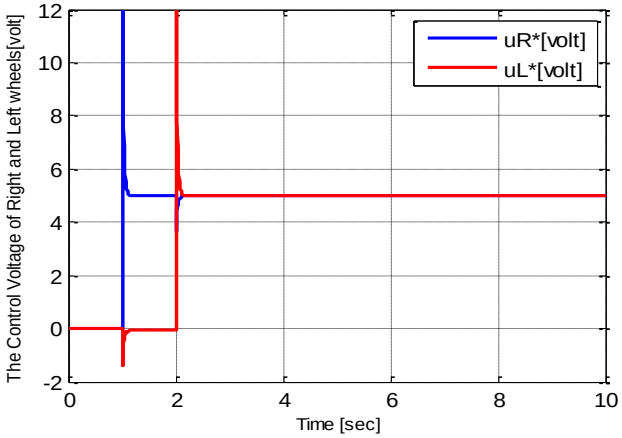
The kinematics and dynamics model of the DDMR described in section II are used. The simulation is carried out by tracking different 3 DOFs desired paths with the high and low level control system of the DDMR. The proposed control-navigation system is implemented based on the structures shown in Fig. 5.

The DDMR with optimal low level control has been tested. Fig. 8.a shows the time response of the right and left motor wheels linear velocities and their corresponding control law i.e., actuator voltage control respectively to a step input command simulated using MATLAB/ Simulink.

As seen from Fig. 8(a, b) the dynamic system response for the right and left wheels is high, and they converge to the desired set value exponentially with no overshot in the system response i.e., the equivalent damping ratio of the system is ζ=1. It is noted that the low level dynamic system is exponentially stable and shows high dynamic performance response with the limitation of *±12volt*



a. Time Response for step commend



b. Motors voltages for left and right wheels

Fig.8. Time responses of the right and left wheels velocity

Two experiments were performed and compared for DDMR with adaptive optimal high level control, with several trajectories to examine the robot dynamic performance, one was implemented utilizing only the onboard sensors data, and the other used EKF assisted fusion sensors method as navigation system considering different system uncertainty.

A. Square Trajectory Tracking:

This section demonstrates the mobile robot control performance to a square path with a side length of 1 meter. Fig. 9 shows the system dynamic response for the high-level control. As seen in Fig 9 that the system exhibited advancement trajectory tracking with a high degree of localization belief has explained by Fig. 10.

As observed in Fig. 10 that the mobile robot system control in the high-level is very certain in its location i.e., where I am, since the degree of belief is high which is also very convincing indicator since the trajectory tracking error is negligible.

Fig. 11 shows the mobile robot speed response, as seen in Fig. 11 the robot moved with maximum speed value 0.14 $m.sec^{-1}$

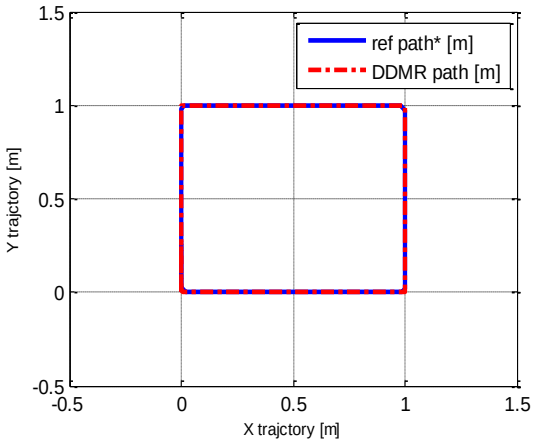


Fig.9. Square Trajectory EKF sensor fusion performance

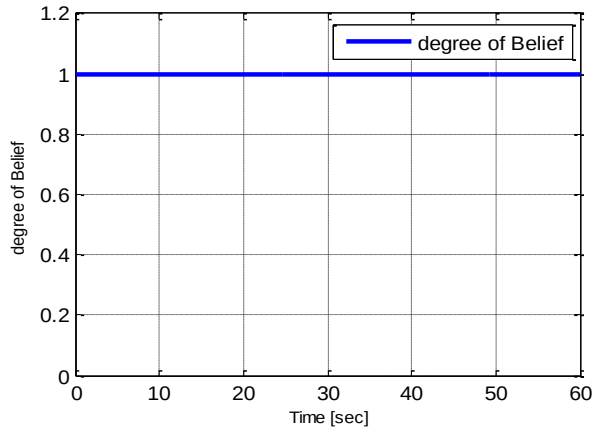


Fig.10. The Degree of Belief for System Navigation

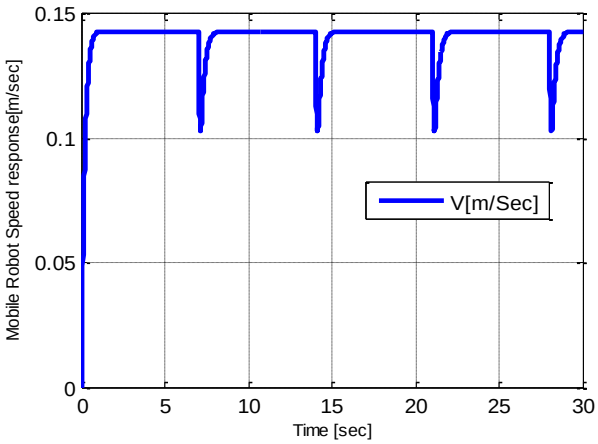


Fig.11. Speed Response of Mobile Robot

B. Eight Shape Trajectory Tracking:

This section provides two case simulation results for 8-shape. These are simulation without system uncertainty and with system uncertainty. As seen in Fig. 12, the robot system tracks correctly the system the desired path (blue) and (red) real trajectory.

The result, in this case, of Fig. 12, was obtained assuming the system is certain and there is no need to use any correction technique to correct the trajectory tracking.

It is seen that the system exhibited a very good convergence to the desired posture of DDMR. The same system is now tested presuming such uncertainty the robot model. It is to kindly remind the reader that the system is to work with no correction technique.

As seen in Fig. 13 the robot dynamics showed a remarkable divergence in the system trajectory tracking (see 'black' reference trajectory and 'blue' real trajectory).

Fig. 14 shows the robot's dynamic performance for an 8-shape with presence of uncertainty in the mobile robot, but the control system was supported with a correction technique for posture correction ('blue' is the reference trajectory, and 'cyan' the real trajectory).

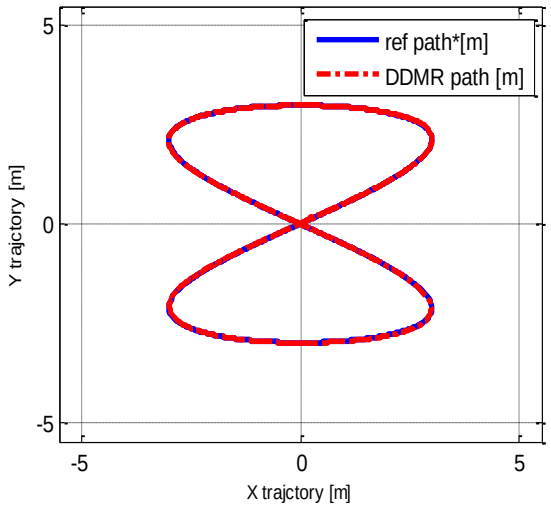


Fig.12. DDMR Trajectory Tracking with Adaptive Optimal Control.

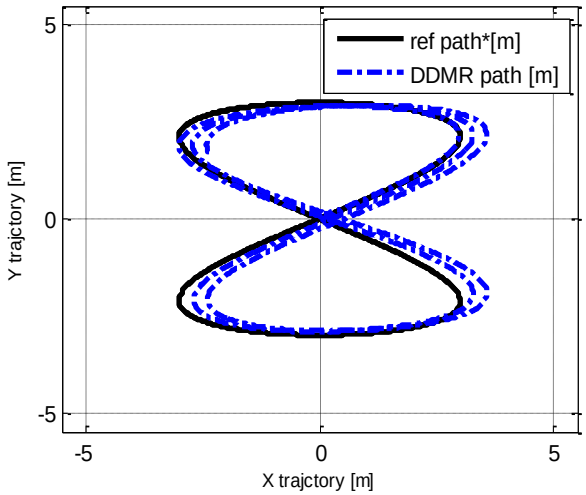


Fig.13. The Accumulated Divergence in the DDMR's Posture with Time

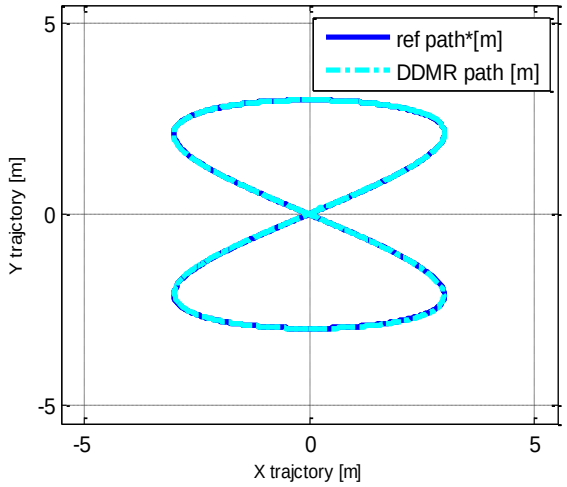


Fig.14. Path Correction based EKF Fusion Sensors Algorithm.

Fig. 15 shows that the proposed Control-Navigation system enhances the robot's belief at different points in time. The solid line displays the actions, and the ellipsoids represent the uncertainty effects on the robot's dynamic performance for the trajectory.

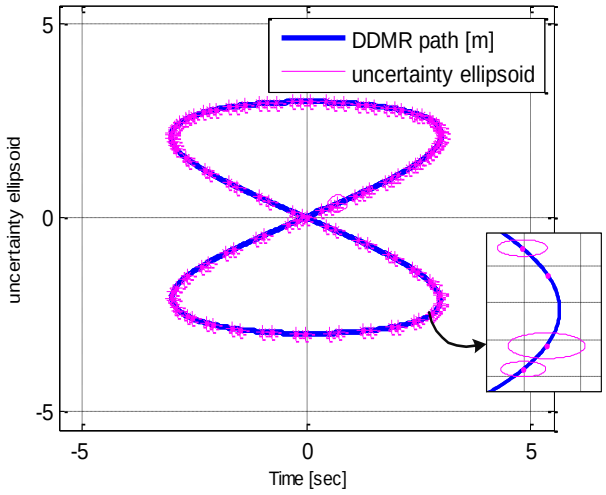


Fig.15. The Posture Believe's DDMR for 8 Shape.

C. *Flower Shape Trajectory Tracking:*

More complex trajectory is used to examine the robot dynamic performance without the presence of uncertainty and with uncertainty. The system dynamic performance is shown in Figures 16, 17, 18 and 19. As noticed in Fig. 19 that the system showed high dynamic performance control for which is the system is capable to correctly track the desired path and even ensure high degree of belief for the robot posture.

It can be observed that better performance control in real time has been obtained by integrating the control system (Low and high levels) with EKF fusion sensors approach to make the control system sensitive to the effects of the environment and able to eliminate the uncertainty effect.

VI. CONCLUSION

This paper has proposed a newly developed approach to enhance the dynamic response of a differential drive mobile robot with the presence of huge system uncertainty. The developed approach has considered a low-level system control developed on the joint space level (the actuators control level) and on the Cartesian space level control i.e., high-level control. For low-level control optimal type controllers (optimal and integral optimal) have been proposed for the actuator velocity control. It has been shown that the dynamic control response of the actuators is high and robust to system uncertainty.

In the high-level control, the system has been incorporated with Extended Kalman Filter (EKF) to estimate, as accurate as possible, the mobile robot posture. Simulation results obtained from the developed control system have shown that a large divergence was due to occur because of system uncertainty, which may lead to erroneous system localization as well as exhibited low degree of belief in the robot location. The proposed control approach in high-level domain overcomes the problem of robot's posture mismatch and tries to correct and compensate the trajectory tracking which may happen due to uncertainty.

The interactive navigation process with the degree of belief, which reflects how the present robot's posture is close to the desired one. The validation of the proposed approach for low-level and high-level control has been confirmed through intensive simulation results obtained for different cases in which the robot has been imposed with uncertainty. As explained the proposed technique has given very encouraging results and has provided very good target tracking with high degree of belief for the robot localization. This work presented in this paper is basis for future. It is to be used for real time implementation for Mobile robot localization system control on the high and low levels identification e.g. UKF in real time utilizing **dSPACE** system board.

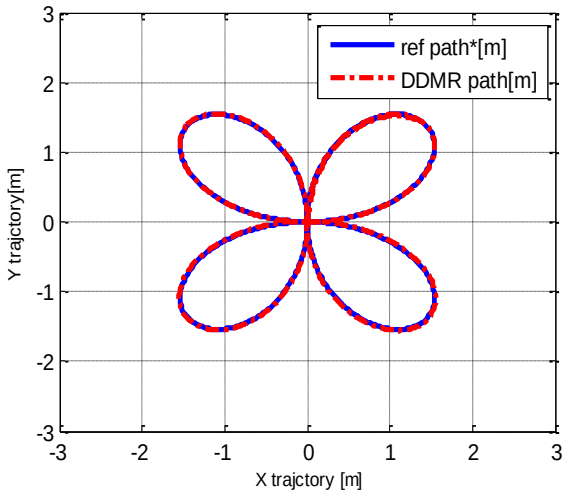


Fig.16. DDMR Trajectory Tracking with Adaptive Optimal Control

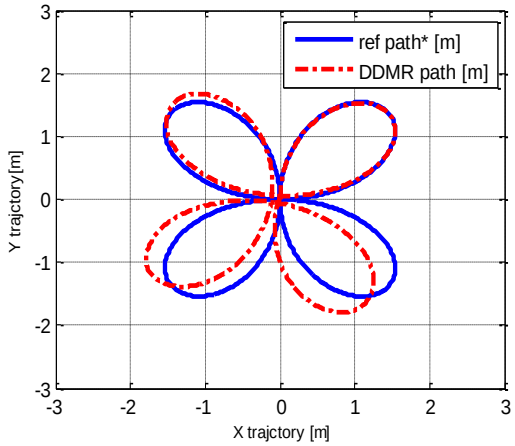


Fig.17. The accumulated divergence in the DDMR's posture with time.

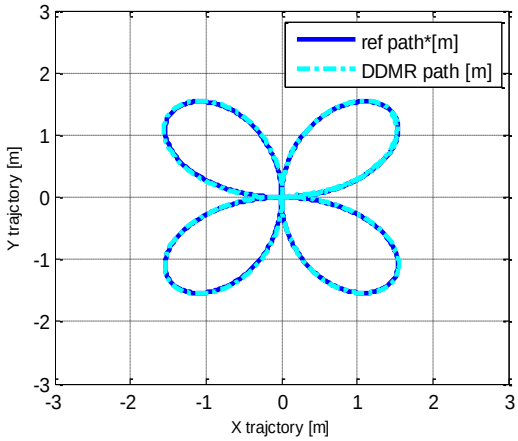


Fig.18. Path Correction based EKF Fusion Sensors Algorithm.

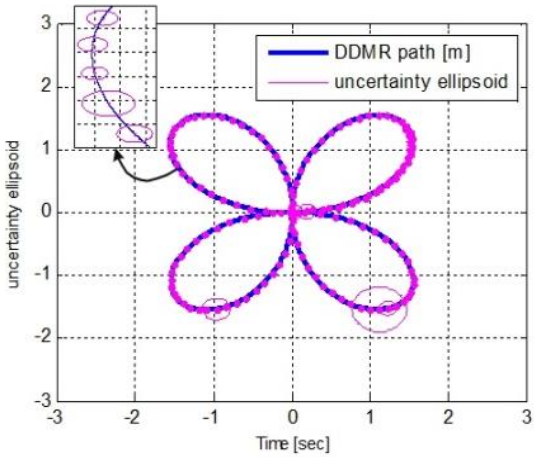


Fig.19. The Posture Believe's DDMR for flower shape.

ACKNOWLEDGEMENT

The authors would like to thank the Syrian Society for Scientific Research (SSSR) for the financial support provided to cover SAI conference registration fee.

REFERENCES

[1] Q. Meng, Y. Sun and Z. Cao "Adaptive Extended Kalman Filter (AEKF)-Based Mobile Robot Localization Using Sonar," Robotica (2000), United Kingdom, Cambridge University Press, vol. 18, pp. 459-473, (December 1999).

[2] Y. Liu, Navigation and Control of Mobile Robot Using Sensor Fusion, Robot Vision, Ales Ude (Ed.), InTech, Available from: http://www.intechopen.com/books/robot-vision/navigationand-control-of-mobile-robot-using-sensor-fusion, (2010).

[3] S. Panich and N. Afzulpurkar "Mobile Robot Integrated with Gyroscope by using IKF," International Journal of Advanced Robotic Systems, vol. 8, No. 2, pp. 122-136, (2011).

[4] R. Negenborn, "Robot Localization and Kalman Filters On Finding Your Position in a Noisy World", UTRECHT UNIVERSITY (september 2003).

[5] S. Thrun, D. Fox and W. Burgard, "PROBABILISTIC ROBOTICS" Massachusttes Institute of Technology (2006).

[6] P. Jensfelt, "Approaches to Mobile Robot Localization in Indoor Environments", Royal Institute of Technology (KTH), (2001).

[7] F. Kühne,W.F.Lages and Gomes da Silva Jr, "Model Predictive Control of a Mobile Robot using Linearization,", IEEE Latin-American Robotics Symposium, pp. 525-530, (2005).

[8] K. Kozlowski and D. Pazderski, "Modeling and Control of a 4-wheel Skid- Steering Mobile Robot,", Int. J. Appl. Math. Comput. Sci., vol. 14, pp. 477-496, (2004).

[9] R. Dhaouadi and A. Abu Hatab "Dynamic Modelling of Differential-Drive Mobile Robots using Lgrange and Newton-Euler Methodologies: AUnified Framework," Advance in Robotics & Automation, vol. 2, (Autom 2013).

[10] S. Haykin, "Kalman Filtering and Neural Networks", A Wiley-Interscience Publication, (2001).

[11] R. Siegwart and I. R. Nourbakhsh, "Introduction to Autonomous Mobile Robots", The MIT Press,Cambridge, Massachusetts London, England, (2004).

[12] K. Ogata, "Designing Linear Control Systems With MATLAB", Prentice Hall,(1994).

[13] P. Corke, " Robotics, Vision and Control Fundamental Algorithms in MATLAB", Springer Tracts in Advanced Robtics, vol 73, (2011).

[14] J.P. Norton, "An Introduction to Identification", Harcourt Brace Jovanovich, Publishers,(1986)

[15] P.S.Maybeck,"Stochastic models, estimation, and control", Harcourt Brace Jovanovich, Publishers,(,vol 1, (1982)

[16] P.S.Maybeck,"Stochastic models, estimation, and control", Harcourt Brace Jovanovich, Publishers,(,vol 2, (1982)

[17] P.S.Maybeck,"Stochastic models, estimation, and control", Harcourt Brace Jovanovich, Publishers,(,vol 3, (1982)

New Hybrid (SVMs-CSOA) Architecture for classifying Electrocardiograms Signals

Assist. Prof. Majida Ali Abed
College of Computers Sciences & Mathematics,
University of Tikrit, Tikrit, Iraq

Assist. Prof. Dr. Hamid Ali Abed Alasad
Computers Sciences Department, Education for Pure Science College, University of Basra, Basra, Iraq

***Abstract*—a medical test that provides diagnostic relevant information of the heart activity is obtained by means of an ElectroCardioGram (ECG). Many heart diseases can be found by analyzing ECG because this method with moral performance is very helpful for shaping human heart status. Support Vector Machines (SVM) has been widely applied in classification. In this paper we present the SVM parameter optimization approach using novel metaheuristic for evolutionary optimization algorithms is Cat Swarm Optimization Algorithm (CSOA). The results obtained assess the feasibility of new hybrid (SVMs - CSOA) architecture and demonstrate an improvement in terms of accuracy.**

Keywords—Electrocardiograms (ECG); classification; support vector machine; Cat Swarm Optimization

I. INTRODUCTION

In recent years, the automatic classification of Electrocardiogram (ECG) signals has received great attention from the biomedical engineering community. Electrocardiography is an important tool in diagnosing the condition of the heart. The electrocardiogram (ECG) is an electrical signal that is generated during the activities of the heart. ECG provides useful information about the functional status of the heart. ECG analysis is an efficient way of diagnosing the abnormal state of the heart. The methods used to diagnose cardiac conditions require technical knowledge, such as that of a physician. Recently, with the development of computer technology many automatic diagnosis methods have been proposed for ECG analysis, including the standard ECG, Blood and Urine tests, Holter Monitoring, Electro-Physiology Studies (EPS), Event Recorder, an echo-cardiogram, Chest X-Ray, Tilt-table test. Using ECG is a common and the best way for diagnosing arrhythmias [1].

Support Vector Machine (SVM) classifier has been successfully applied to the problem of the automatic classification of ECG signal [2]. For obtaining satisfactory predictive classification accuracy, we can use various SVM kernel parameters. Therefore, it needs to be a convenient and efficient kernel parameter setting method [3]. In this paper, we present the SVM parameter optimization approach based on Cat swarm optimization Algorithm (CSOA). Selection of the parameters is an important factor affecting the performance of the SVM. The CSOA is applied to estimate the optimal parameters of SVM classifier

The next section describes the related work. To describe the Electrocardiograms (ECG) signal, Section III presents the ECG signal as five important component basic activities. Section IV is devoted to the pattern recognition Support Vector Machines (SVMs). In section V, we present a stochastic optimization algorithm which is called Cat swarm optimization (CSOA) to improve classification accuracy. In Section VI we describe our proposed architecture in detail. In Section VII we expose the results obtained from our experiments. Finally Section VIII presents the conclusions.

II. RELATED WORK

Several algorithms have been developed in the literature to improve the detection and classification of Electrocardiogram (ECG) Signals. Better performance depends on features and the methods of classification, early approaches are mostly based on Artificial Neural Networks (ANNs) [4], which have been used in a great number of medical diagnostic decision systems [5, 6]. The most popular neural network is Multi-Layer Perceptron (MLP), fuzzy logic [7], Support Vector Machines SVMs used for Examining feature extraction techniques for ECG classification, Ant colony optimization, k-nearest neighbor [8]. Little research exists that investigates the feasibility of using Evolutionary classifiers for Electrocardiogram (ECG) detection. However, the known methods for Electrocardiograms (ECG) detection mostly focus on differentiating between different Electrocardiogram (ECG) types. Support Vector Machine (SVM) has been used for principal classification but the features' set is evolved through a genetic search. Similarly, some work has done on Electrocardiograms (ECG) signals classification [9].

III. ELECTROCARDIOGRAM (ECG) SIGNAL

Electrocardiogram (ECG) is a non-invasive method of measuring the electrical properties of the heart. It is used to measure the electrical activity and is a common way for detecting heart arrhythmia. These electrical changes that spread through the heart provide information about the functional aspects of the heart and of the cardiovascular system. We must save the ECG signal in order to determine different types of the heart disease. The ECG signal is recorded to identify the change in heartbeat. Every ECG signal consists of five important component basic activities (P, T, Q, S, R waves). All these characteristic points should be detected. QRS wave group complex is ventricular depolarization which has the biggest slop in ECG [10].

- P wave is a trial depolarization
- R is the distance between the peaks of QRS current and previous pulse

- Q is first point before R, which slops less than zero.
- S is first point after R which slops less than zero.
- T wave is ventricular repolarization and equal (R-(Q+S)/2)*0.3+(Q+S)/2

These five important component basic activities (P, T, Q, S, R waves) which are used for the interpretation of the ECG show in Figure (1).

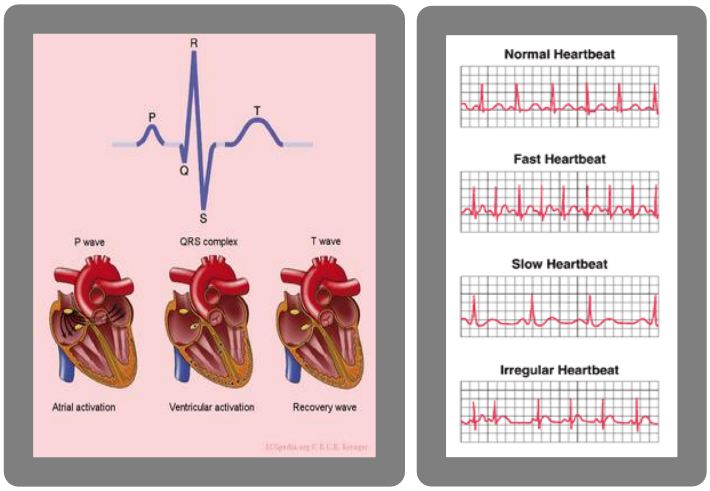


Fig.1. Standard ECG beat, ECG wave form and Process of Heart beat recoding by ECG Signal.

One cycle of ECG signal consists of P-QRS-T wave, six types of beats including Normal Beat, Premature Ventricular Contraction (PVC), Fusion of Ventricular and Normal Beat (F), Atrial Premature Beat (A), Right Bundle Branch Block Beat (R) and Fusion of Paced and Normal Beat (f). Figure(2) shows the ECG signal processing flow. Features such as energy and entropy of the ECG signals, were then extracted from these decomposed signals as feature vectors [11]. The output of Electrocardiography is a graph of two-dimensional plot, the x-axis represents time in seconds and the y-axis represents signal voltage in milli-volts. Conventional ECG machines print their output on grid graph papers, each square grid is $1mm^2$.

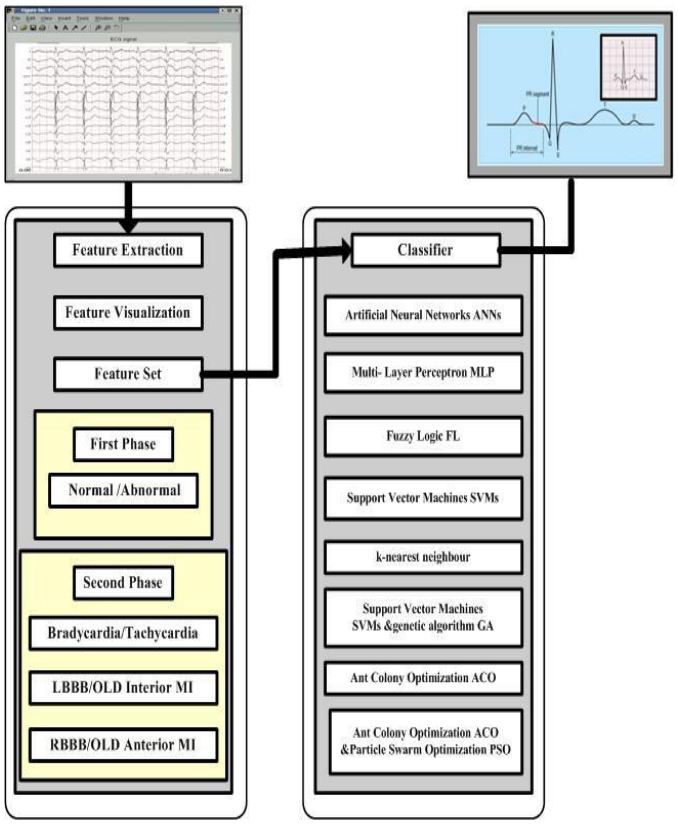


Fig.2. Electrocardiograms ECG Analysis Framework.

IV. Support Vector Machines (SVMs)

The Support Vector Machines (SVMs) possess great potential and superior performance as has been shown in many previous researches for efficiently training linear learning machines in kernel-induced feature spaces. SVMs are pattern recognizers that classify data without making any assumptions about the fundamental process by which the explanations were granted. In SVM data can be seen in the form of P-dimensional vector [12]. SVM performs classification tasks by constructing Optimal Separating Hyper-planes (OSH). OSH maximizes the margin between the two nearest data points belonging to two separate classes.

So the following inequality is valid for all input data: The SVMs use hyper- planes to separate the different classes. The SVM approach seeks to find the optimal separating hyper-planes between classes. The hyper- plane is constructed so as to maximize a measure of the 'margin' between classes. Figure (3) The Limitations of SVM are the performance of SVMs which largely depend on the choice of kernels, the choice of kernel functions, which are well suited to the specific problem, is very difficult, speed and size are other problems of SVMs both in training and testing. In terms of running time, SVMs are slower than other neural networks for a similar generalization and performance. For given training data, it is believed that SVS will perform well when the patterns to be classified are not separable and the training data is noisy.

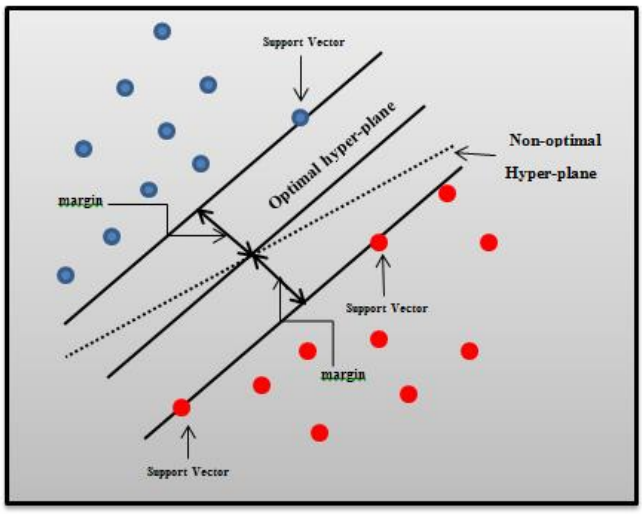


Fig.3. SVM Model Optimal hyper-plane.

The goal of SVM is to minimize the expectation of the output of sample error. SVM map is a given set of binary labelled of each training sample to a high dimensional feature space and separate the two classes of sample is available with a maximum margin of hyper-plane. SVM algorithm seeks to maximize the margin around a hyper-plane, which separates a positive class from a negative class as shown in equations (1), (2) and (3), [13].

$$f(x) = \left(w.\Phi(x)\right) + b \tag{1}$$

$$\mathrm{R_{SVM}(c)} = \mathrm{c}\frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{L_\varepsilon(y_{i.}y_i^\Lambda)} + \frac{1}{2}\mathrm{w^T.w} \tag{2}$$

$$\mathrm{L_\varepsilon(y_{i.}y_i^\Lambda)} = |\mathrm{y_{i.}} - \mathrm{y_i^\Lambda}| - \varepsilon \quad |\mathrm{y_{i.}} - \mathrm{y_i^\Lambda}| \geq \varepsilon$$

$$= 0 \qquad otherwise$$

$$^{\Lambda}\mathrm{y=f(x)=} \sum_{\mathrm{i=1}}^{\mathrm{N}} (\alpha_\mathrm{i} - \alpha_\mathrm{i}^*)\mathrm{k(x_i, x)} + \mathrm{b} \tag{3}$$

$$\mathrm{k(x_i, x)} = \exp\left(\frac{-1}{\delta^2}\left(\mathrm{x_i} - \mathrm{x_j}\right)^2\right)$$

Where

$\phi(x)$: is the non-linearity high dimension feature space which mapped from the input space *x.*
w : is the modifiable model parameter.
b : is the threshold value.
w and *b :* are estimated by minimizing.
$\propto_i \ and \ \propto_i^*$: are the Lagrange multipliers, which are positive real constants. The data points corresponding to non-zero value for $(\propto_i - \propto_i^*)$ are called support vectors.
$k(x_i, x)$: is the inner product kernel function (Gaussian Kernel).
δ^2 : is the Gaussian kernel factor (the width of the kernel function).
C: positive real constant controls the trade-off between training error and generalization ability and its value is chosen by means of a validation set.
ε: parameter for SVMs.

V. Cat Swarm Optimization Algorithm

Many researchers recently found algorithm optimization techniques mimic animal behavior. A new algorithm introduced by Chu, Tsai and Pan in 2006 was named Cat swarm optimization Algorithm (CSOA) [14]. This algorithm is a kind of swarm intelligence that is based on stochastic optimization inspired by social behavior, as well as contributing to computer engineering applications. The problem faced in few researches is how to develop Cat Swarm Optimization algorithm that can be used in data mining, especially for the case of classification, but rarely or never used until now in pattern matching Problems. CSOA has a number of advantages in pattern matching problems of optimization compared to previous techniques such as Genetic algorithm (GA), Ant Colony Optimization Algorithm (ACOA), Particle Swarm Optimization (PSO) and Binary Particle Swarm Optimization (BPSO). With CSO algorithm development, expected produce a faster time and has a better accuracy rate compared to existing algorithms [15].In the CSOA, a set of cat behaviour in two different modes: Searching Mode (TM) and the Tracing Mode (SM) are used to resolve the optimization problem. The first mode of the CSO is to determine how many cats will be used in iteration, then use the cat in the CSOA to resolve the problem. CSOA is an evolutionary optimization algorithm is modelled on two major behavioral traits of cats. These behaviors are termed as seeking mode (Cats move slowly when resting but being alert) and tracing mode (Cats move slowly when resting). In seeking mode, we define four important factors: seeking memory pool (SMP), seeking range of the selected dimension (SRD) (to find a range of selected dimensions), counts of dimension to change (CDC) (to calculate dimensions will change), and self-position considering (SPC) (to consider the position). The tracing mode of CSO algorithm that describes the cat is being followed the lead of the target. Once a cat goes into tracing mode, it moves according to its' own speed for each dimension [16]. The detailed descriptions of these modes are given in the general process of Cat Swarm Optimization Algorithm CSOA in Figure (4). It is explain the two modes of CSO algorithm, the Mixture Ratio (MR) indicates the rate of mixing the seeking mode and the tracing mode. The process of Cat Swarm Optimization Algorithm CSOA is described as follows:

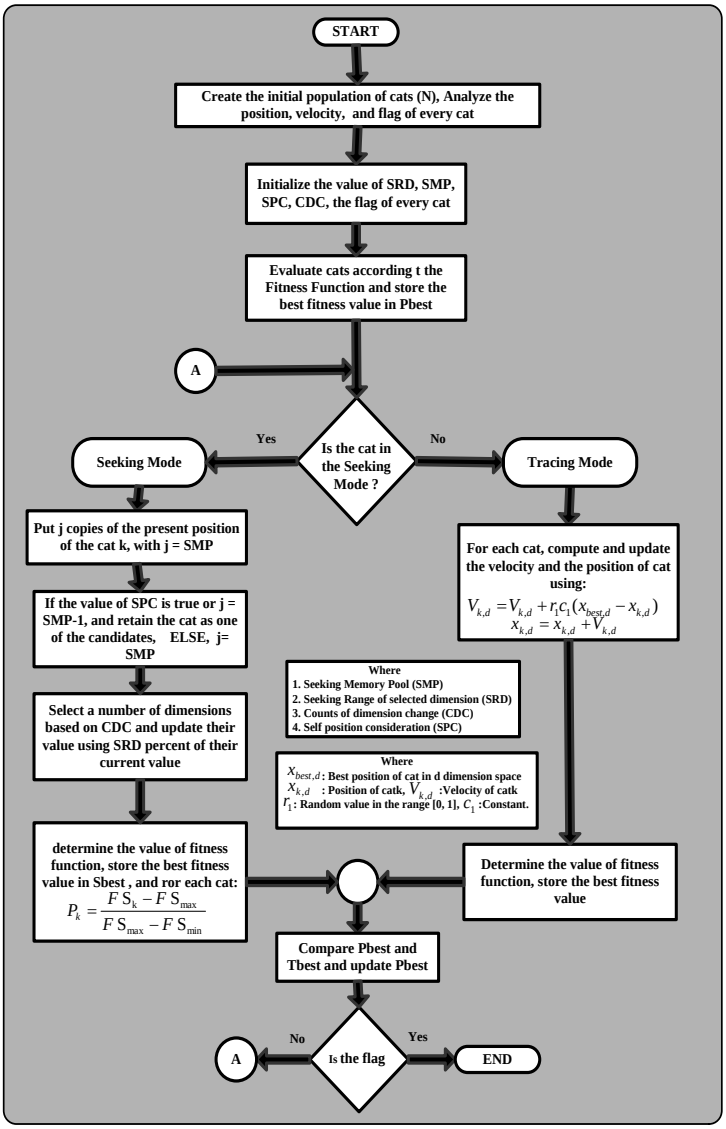


Fig.4. Figure 4 The general process of Cat Swarm Optimization Algorithm CSOA.

VI. PROPOSED SYSTEM

The Electrocardiograms ECG electrodes convert heart signals into an electrical signal ranging from 1mV to 5 mV. Every ECG signal has five distinct points (P, Q, R, S and T wave). Figure (5) below explained the Flowchart of CSOA based parameter optimization for SVM classifier of our proposed system which is composed of three stages:

- Pre-processing
- Feature Extraction
- Classification using SVMs

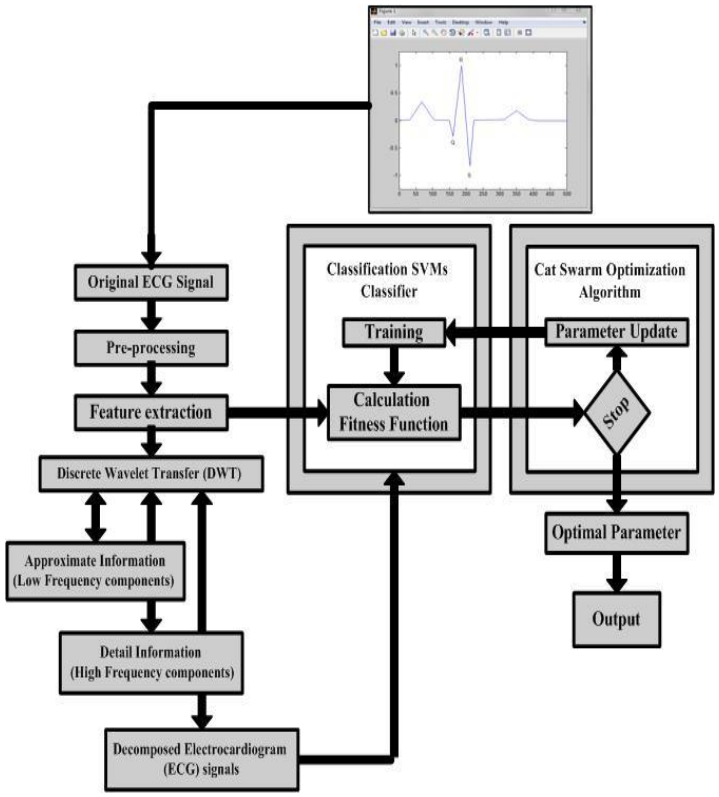


Fig.5. Flowchart of our proposed system CSOA based parameter optimization for SVM classifier.

A. Pre-processing

The noise of Electrocardiograms ECG signal contains power line interference, baseline wandering effect, and muscle noise. The common noise of raw ECG signals is shown in Figure (6).

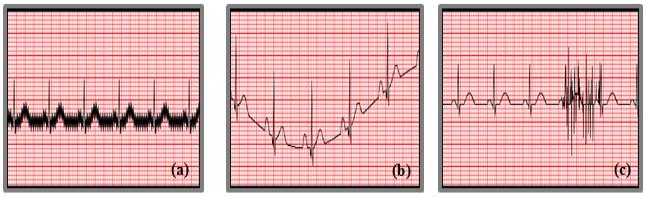


Fig.6. Examples of different ECG signal noise (a) power line interference, (b) baseline wandering effect(c) muscle noise.

The power-line interference is caused by the electromagnetic field interfering with the ECG equipment cables. Baseline wandering effect causes the entire ECG signal to shift up or down from the normal base at zero y-axis,. A muscle noise is an interference voltage generated whenever a patient contracts a body muscle. These micro-voltages are detected by the electrode and shown in the signal as noise.

To remove all noise from raw ECG signals, we can implement conservative filters. A high-pass filter with 0.5Hz cutoff frequency can be used to remove unwanted low-frequency components of baseline wandering effect. The power-line noise is filtered away by a notch filter, and the muscle noise effect can be removed by a time-varying low-pass filter. in the pretreatment of ECG wave is completed based on a high-pass filter with 0.7Hz and low-pass filter with 100Hz, Figures (7) (a) and (b) [17].

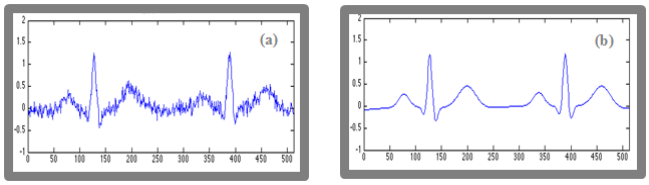


Fig.7. (a) ECG signal before de-noising, (b) ECG signal after de-noising.

In order to extract and use the feature of the ECG waveform in the classifier design stage of ECG signal analysis, the noise elimination, baseline wanders removal and peak detection (P, Q, R, S and T) [15].

B. Feature extraction

Feature extraction plays an important role in any classification task. The purpose of feature extraction is to select and retain relevant information from the original signals.The feature extraction in this paper uses the Discrete Wavelet Transform (DWT- Daubechies16 (Db16)) [18]. ECG signals were decomposed to the approximate information called low frequency components and detailed information called high frequency components show in Figure (1). Decomposition of the signal is done up to many steps (eight steps). If these steps are high, the low-frequency components of the original signal are better conserved. Low frequency band of the ECG signal is used for the detection of QRS, T, and P waves. For each feature shown in table (1) are calculated. Execution of ECG classifier is done by using SVMs distribution estimation (one-class SVMs).

C. ECG Classification

ECG classification system based on Support Vector Machines SVMs One cycle of ECG signal consists of P-QRS-T wave, six types of beats including Normal sinusrhythm (N), a trial premature beat (A),Ventricular premature beat (V), Right Bundle branch block (RB), Left Bundle branch block (LB), and Paced beat (/).

The ECG signals were first decomposed with the discrete wavelet transform (DWT) after which feature vectors were extracted[19]. These feature vectors were used to classify the signal. In our proposed system we take ten features (QRS Complex ,P-R Segment ,P-R Interval ,S-T segment ,Q-T Interval , R-R Interval, P-P Interval-R and P-P Similarity ,R-R Interval Variance, Heart Beat), they were saved as separate vectors and extracted for each heart beat automatically, then we calculated the difference position between two separate vectors as feature values after detection of the position of Q, R, S, T start, T end, P start and P end. The description of these features has been explained in table (1) below for each heartbeat. The start of the QRS complex was defined as the beginning of each beat and normal beats which occurred immediately before or after abnormal beats were removed[20].

TABLE I. TEN FEATURES DESCRIPTION USED FOR EACH HEART BEAT AUTOMATICALLY.

No.	Feature name	Description of Feature	Pos position of feature in the ECG signal
1	QRS Complex	Pos(S) - Pos(Q)	
2	P-R Segment	Pos(Q) - Pos(Pend)	
3	P-R Interval	Pos(Q) - Pos(Pstart)	
4	S-T segment	Pos(Tstart) - Pos(S)	
5	Q-T Interval	Pos(Tend) - Pos(Q)	
6	R-R Interval	Pos(Rnext) - Pos(R)	
7	P-P Interval	Pos(Pnext) - Pos(P)	
8	R-R and P-P Similarity	ABS((R-R Interval)- (P-P Interval))	
9	R-R Interval Variance	VAR(R-R Interval)	
10	Heart Beat	60/(R-R Interval)	

D. Parameter optimization based on CSOA

To determine the optimized parameter using the CSOA. Randomly generate *N* solution sets and velocities with *D*-dimensional space, represented the following parameters of cat swarm optimization algorithm explain in the table (2).

TABLE II. PARAMETERS OF CAT SWARM OPTIMIZATION ALGORITHM.

No.	parameter	Description
1	SMP	seeking memory pool
2	SRD	seeking range of the selected dimension
3	CDC	counts of dimension to change
4	MR	mixture ratio
5	NBS	number of best solution sets
6	MR Best	mutation rate for best solution sets
7	NTM	number of trying mutation

We present an CSOA to search the optimal penalty coefficient C_{best} , insensitive loss coefficient ε_{best} and the width of kernel function δ_{best} in the SVM forecast parameters space (C,ε,δ). Create N cats, regulate N and dusting the adjusted N cats into the 3 dimensional SVM forecast parameter space (C,ε,δ) evenly by the Even Distribution Process and divide the adjusted N cats into G groups. Randomly generate the Velocities for each dimension V_C, $V\varepsilon$ and $V_\delta$of each cat. This should be in the predefined range. Set the Motion Flag of each cat to make them move into the Parallel Tracing Mode or the Seeking Mode according to the predefined value of MR, where $MR\in[0,1]$For each cat, take location coordinates (C_i, ε_i ,δ_i) into the Fitness Function of SVM, calculate the fitness values respectively, and record the Location coordinates and the fitness values [21-23].

VII. EXPERIMENTS AND RESULTS

This section describes the experimental setup of dataset based on MATLAB. We description the dataset and their training and testing data beats are explained in the two paragraphs below:

- Dataset Description.
- Results and Discussion.

A. Dataset Description

Our experiment conducted on the basis of ECG data from the Public teaching hospital in Tikrit, Heart catheterization suite database, were chosen from the recording of (30)

patients, which matched the following files, Table (3) displays the datasets employed in our experiment.

TABLE III. DISPLAYS THE DATASETS EMPLOYED IN OUR EXPERIMENT.

No. of patient	No. of file	No. of patient	No. of file
1	0100	16	0205
2	0102	17	0206
3	0103	18	0207
4	0104	19	0208
5	0105	20	0209
6	0106	21	0210
7	0107	22	0211
8	0108	23	0212
9	0109	24	0213
10	0110	25	0214
11	0200	26	0215
12	0201	27	0216
13	0202	28	0217
14	0203	29	0218
15	0204	30	0219

B. Results and Discussion

Our proposed system followed the datasets of 30 patients, the number of files for every penitent explained in table (3) which are chosen from Public teaching hospital in Tikrit, to measure the classification accuracy of the proposed hybrid CSOA-SVM method. The measured Electrocardiograms ECG signal refers to succeeding classes of beats:

- Normal sinusrhythm (N)
- Trial premature beat (A)
- Ventricular premature beat (V)
- Right Bundle branch block (RB)
- Left Bundle branch block (LB)
- Paced beat (/).

Our experiment in order to deliver for the classification procedure the two following kinds of features are espoused:

1) ECG morphology features
2) Ten ECG temporal features

The ten features are (QRS Complex, P-R Segment, P-R Interval, S-T segment, Q-T Interval, R-R Interval, P-P Interval-R and P-P Similarity, R-R Interval Variance, Heart Beat). Then, extracting the following ten Temporal features of interest, normalized to the similar length the period of the segmented ECG cycles according to the process reported. To this end, the mean beat length was chosen as the standardized length, which was represented by 300 evenly distributed samples. Accordingly, the total number of morphology and sequential features equals 310 for each beat.

In our experiments, ten different tribunals are performed, each with a new set of randomly selected training beats, while the test set was kept unbothered. The results of these ten tribunals attained on the test set were thus averaged. The thorough amount of training and test beats are described for each class in Figure (8).

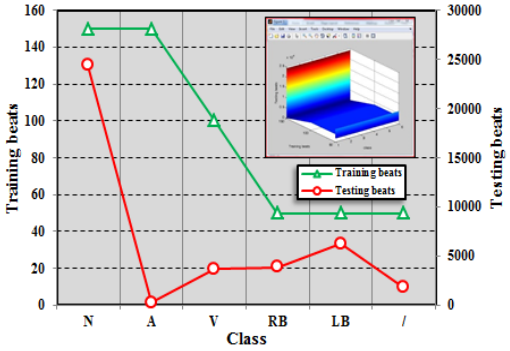


Fig.8. Numbers of Training and Testing Beats used in our Experiments.

Classification presentation was evaluated in terms of two measures, which are:

1) Overall Accuracy (OA), which is the percentage of correctly classified Electrocardiograms (ECG) signal among all the beats considered.

2) Average Accuracy (AA), which is the average over the classification accuracies obtained for the different classes.

Overall Accuracy (OA), Average (AA), and succeeding classes of beats (Normal sinusrhythm (N), Trial premature beat (A), Ventricular premature beat (V), Right Bundle branch block (RB), Left Bundle branch block (LB), Paced beat (/)) Attained on the Test Beats with the Different Investigated Classifiers with a total Number of 550 Training Beats. The accuracy of the intentional system of hybrid CSOA-SVMs may have high accuracy when evaluated with the standard SVMs. The hybrid CSOA-SVMs has high value in all proposed systems. Figure (9) explains the result of our experiment.

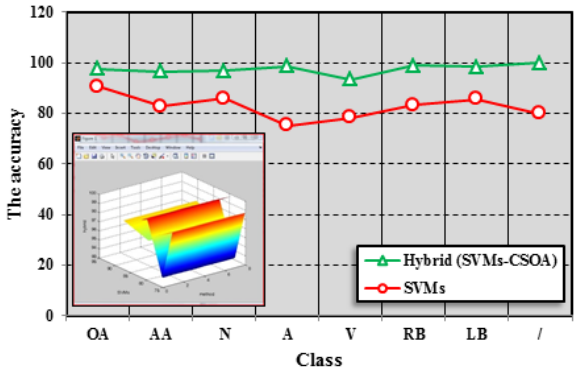


Fig.9. The hybrid CSOA-SVMs the result of our experiment.

VIII. CONCLUSION

The natural world secretes many characteristics of different creatures, and all of them have some exceptional behaviors or features to keep them subsist in our paper, CSOA-SVMs approach is proposed for an automatic Electrocardiograms ECG signal classification. This approach presents methods for improving SVMs performance in two aspects: feature selection and parameter optimization.

The new method proposed in this paper is the combination of a support vector machine and cat swarm optimization algorithm (CSOA-SVMs). This hybrid system is jointly applied to optimize the feature selection and the SVMs parameters (penalty coefficient C_{best} , insensitive loss coefficient ε_{best} and the width of radial basis function kernel function δ_{best} in the SVM prediction parameters space (C, ε ,δ).

REFERENCES

[1] B. Mohammadzadeh Asl, S.K. Setarehdan, and M. Mohebbi, "Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal," Artificial Intelligence in Medicine, vol. 44, no.1, pp. 51-64, 2008.

[2] Osowski, S., Markiewicz, T., & Tran Hoai, L., "Recognition and classification system of arrhythmia using ensemble of neural networks", Measurement, 41, pp. 610-617, 2008.

[3] Polat, K., Akdemir, B., & Günes, S., "Computer aided diagnosis of ECG data on the least square support vector machine", Digital Signal Processing, 18(1), 2008.

[4] Sarkaleh, Maedeh Kiani; Shahbahrami, Asadollah, "A. Classification of ECG Arrhythmias Using Discrete Wavelet Transform and Neural Networks", International Journal of Computer Science, Engineering and Applications, Vol. 2, Issue 1, p1 ,2012.

[5] Javadi M, Ebrahimpoor R, "improving ECG classification accuracy using an ensemble of neural network modules", PLos ONE 6(10).journal.pone.0024386.

[6] Rahat Abbas, Wajid Aziz, and Muhammad Arif, "Prediction of Ventricular Tachyarrhythmia in Electrocardiograph Signals using Neural Network and Modified Nearest Neighbour Method", Proceedings of International Students Conference on Engineering Sciences and Technology (SCONEST), 2004.

[7] Moein, S.T., Advances in Computational Biology: "A MLP Neural Network for ECG Noise Removal Based on Kalman Filter", In: Arabnia, H.R. (ed.). Springer, Heidelberg, 2010.

[8] M. Korürek and A. Nizam, "Clustering MIT-BIH arrhythmias with ant colony optimization using time domain and PCA compressed wavelet coefficients," Digital Signal Processing, vol. 20, no. 4, pp. 1050–1060, 2010.

[9] M. J. del Jesus, F. Hoffmann, L. Junco, L. Sánchez. "Induction of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms", IEEE Transactions on Fuzzy Systems 12:3, pp 296-308, 2004.

[10] Bortolan G, Willems JL, "Diagnostic ECG classification based on neural networks", Journal of Electrocardiology, Vol. 26, pp. 75-79, 1994.

[11] Kutlu Y, Kuntalp D., "A multi-stage automatic arrhythmia recognition and classification system". Computers in Biology and Medicine,Vol41(1), pp. 37–45, 2011 .

[12] Wu Q. and Zhou D.-X., "Analysis of Support Vector Machine Classification", International Journal of Computer Analysis and Applications Vol. 8, pp. 99–119, 2006.

[13] Karpagachelvi S, Arthanari M, Sivakumar M., "classification of electrocardiogram signals with support vector machines and extreme learning machine", Neural Computing and Applications, Vol. 21(6), 2012.

[14] S. C. Chu, P. W. Tsai and J. S. Pan, "Cat swarm optimization", proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence, pp. 854-858, ,2006, Guilin, China.

[15] S. C. Chu and P.W. Tsai, " Computational intelligence based on the behavior of cats", International Journal of Innovative Computing, Information and Control, Vol.3,No.(1), 2007.

[16] J. Paul and T. Yusiong, "Optimizing Artificial Neural Networks using Cat Swarm Optimization Algorithm," I. J. Intelligent Systems and Applications, vol. 1. 2013.

[17] S.Paland M. Mitra, "Detection of ECG characteristic points using multi-resolution wavelet analysis based selective coefficient method", Measurement, Vol.43. 2010.

[18] R. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 2001.

[19] W. Jiang and G. Seong Kong, "Block-Based Neural Networks for Personalized ECG Signal Classification," Neural Networks, IEEE Trans. Vol. 18, No. 6, 2007.

[20] D. Liu, "Research on Quantum Neural Network Model and Its Application to ECG Classification", Master's thesis, Nanjing University of Posts and Telecommunications, 2012.

[21] J. A.Nasiri, M. Sabzekar, H. S. Yazdi, M. Naghibzadeh, and B. Naghibzadeh, "Intelligent Arrhythmia Detection Using Genetic Algorithm and Emphatic SVM(ESVM)," in Third UKSim European Symposium On Computer Modeling and Simulation, pp. 112–117, 2009.

[22] Melgani F, Bazi Y., "Classification of Electrocardiogram Signals with Support Vector Machines and Particle Swarm Optimization", IEEE Trans. On Information Technology in Biomedicine, Vol.2, No. (5), 2008.

[23] Kuan-Cheng Lin, Yi-Hung Huang, Jason C. Hung, and Yung-Tso Lin., "Feature Selection and Parameter Optimization of Support Vector Machines Based on Modified Cat Swarm Optimization", International Journal of Distributed Sensor Networks, 2014.