Event-triggered Consensus Control Approach for Guaranteed Finite Time Convergence - *Project Proposal*

Liam Dale[1], n9741283
Dr. Aaron McFadyen[2]

***Executive Summary* – This document outlines the motivations and details for an investigation into event-triggered consensus of multi agent systems guaranteeing finite time convergence. An initial literature review suggests that this strategy effectively uses system resources while having strong performance. The first of two phases aims to establish a foundation by implementing a simple event triggered consensus strategy. The second phase plans to guarantee finite time convergence and design for robustness.**

I. Project Introduction

A century of aerospace innovation is culminating to the development of urban air mobility (UAM), which includes fully automated vertical take-off and landing aircraft (VTOL) for intra-city transportation [1]. When compared to road transport this technology may alleviate congestion and decrease travel time while faring at par or better regarding fuel costs and carbon dioxide emissions [2]. This has piqued interest from many entities including Uber and Airbus, which are already rolling out services on a small scale [3, 4]. Due to its infancy there are still many open questions regarding regulations and technical implementations [5].

Successful UAM realisation will see an increase in vehicle density along major aerial routes and highways which will eclipse the capability of contemporary air traffic management (ATM) systems. Thus, a challenge is posed to provide technologies which autonomously maintain safety and efficiency for dense multi-agent systems (MASs) with networking capabilities [1]. This will require all agents coordinate to observe a collective behaviour.

This project will focus on the topic of consensus, which typically refers to the problem of reaching an agreement among a group despite their initial state trajectories [6]. This is an integral component of group decision making and motion coordination. A typical example is platooning on highways, where vehicles negotiate a collective speed and heading to maintain a safe separation [6]. This project report focusses on decentralised event-triggered consensus as this has practical implications relevant to UAM.

This report is a project proposal, structured as follows. Section II provides a background and literature review to further introduce the topic. Findings inform the research problem, which is outlined within Section III. Finally, section IV suggests a strategy and timeline for implementation.

II. Background/Literature Review

Information sharing between MAS participants is essential for consensus such they may all agree on a common goal [6]. It may occur in a centralised manner, which assumes all agents communicate global knowledge over a fully connected network and solve the problem with a central node. This approach introduces a single point of failure and as communication topologies are rarely fully connected becomes increasingly impractical as the number of agents scales [7]. Alternatively there is a decentralised scheme, where information may only be shared between neighbours and nodes perform individual decision making. This method is pragmatic by virtues of robustness and scalability, however is more complex in structure and organisation [7].

Continuous communication between agents is unrealistic for practical applications, as it is an unnecessary drain on limited resources [8]. Computation, actuator updates and communication consume energy to drain batteries and curtail flight time. Networks constraints such as packet loss, latency and throughput worsen with congestion, endangering performance and stability [8]. Drawbacks are exacerbated as the network size scales. As such, it is important to conservatively transmit information while preserving the overall system performance [9]. Event triggered control is a solution where transmissions are spread sporadically based on current system measurements and an event-triggering threshold. This on-demand strategy significantly reduces transmission frequency while upholding performance [9].

A. *Multi-Agent Consensus*

The network topology plays an integral role in the consensus of a multi-agent system, determining the rate of convergence, negotiated result and whether consensus is possible [6]. The topology is not necessarily fixed, being affected by vehicle motion and communication dropouts. Suppose there are n agents where the communication topology is modelled by a directed graph $G_n \triangleq (V_n, E_n)$. This system must have negotiations arbitrated by a consensus algorithm.

[1] **Liam Dale** is with Science and Engineering Faculty (SEF), Queensland University of Technology (QUT), Australia. This report is in partial fulfilment of EGH400-1 unit assessment requirements and submitted on 06-September-2020. liam.dale@connect.edu.au.
[2] **Dr. Aaron McFadyen** is also with Science and Engineering Faculty (SEF), Queensland University of Technology (QUT), Australia. He is the supervisor for this research project.

The most prevalent continuous-time consensus algorithm is given as [6]

$$\dot{x}_i(t) = -\sum_{j=1}^{n} a_{ij}(t)\left[x_i(t) - x_j(t)\right], \qquad i = 1, \dots, n$$

where $a_{ij}(t)$ is an entry in the associated adjacency matrix. State x denotes the information state, which relates to the consensus variable of interest. The corresponding matrix form is

$$\dot{x}_i(t) = -L_n(t)x(t)$$

where $L_n(t)$ corresponds to the G_n associated Laplacian matrix. Consensus is achieved when for each initial state $x_i(0)$ and for all $i, j = 1, \dots, n$ the difference of states $\left|x_i(t) - x_j(t)\right| \to 0$ as $t \to \infty$.

The continuous time algorithm is discretised with a difference equation [6]

$$x_i[k+1] = \sum_{j=1}^{n} d_{ij}[k]x_j[k], \qquad i = 1, \dots, n$$

where $d_{ij}[k]$ is an entry in the associated row-stochastic matrix D, and k denotes a communication event. Information states are held constant between triggers. The corresponding matrix form is

$$x[k+1] = D[k]x[k]$$

Similarly, consensus is achieved when for each initial state $x_i[0]$ and for all $i, j = 1, \dots, n$ the difference of states $\left|x_i[k] - x_j[k]\right| \to 0$ as $k \to \infty$.

For both continuous and discrete time with invariant topologies and constant a_{ij} gains consensus is achieved when the directed topology has a directed spanning tree, or the undirected topology is connected. In these scenarios the Laplacian matrix eigenvalues $\lambda_i(L_p)$ are positive semidefinite, starting from zero and increasing. As such, non-zero eigenvalues $\lambda_i(-L_p)$ are all negative. The second smallest $\lambda_2(L_p)$ is the algebraic connectivity and defines the asymptotic convergence rate [6].

These algorithms force x_i to be driven towards the information state of its neighbours. They do not specify a goal state to reach – however this is achievable through extending the algorithm to include a reference model [6]. Equilibrium is determined by vehicles which are the root of a directed spanning tree, equal to a weighted average of their initial states. This property gives rise to leaderless, and leader-follower strategies. For undirected connected graphs this encompasses all connected nodes giving average consensus [6].

With controllers being connected over a network, the consensus should be robust to wireless channel constraints and stochasticity [7]. Issues such as fading, variable latency, packet misses and quantisation cause the system and controller inputs to be an approximation of measured values. These issues have the potential to degrade performance and stability [7], and alongside network topology changes are not addressed in the general algorithm mentioned.

B. *Consensus for Single-Integrator Dynamics*

Consider agents where the information state dynamics are given as the control input [6].

$$\dot{\xi} = u_i(t), \qquad i = 1, \dots, n$$

The fundamental consensus algorithms are then altered as follows

$$\mathrm{u_i(t)} = -\sum_{j=1}^{n} a_{ij}(t)\left[\xi_\mathrm{i} - \xi_j\right], \qquad i = 1, \dots, n$$

$$\xi_i[k+1] = \sum_{j=1}^{n} d_{ij}[k]\xi_j[k], \qquad i = 1, \dots, n$$

and in matrix forms

$$\dot{\xi} = -|L_n(t) \otimes I_m|\xi$$

$$\xi[k+1] = -|D_n[k] \otimes I_m|\xi[k]$$

The concepts for stability and convergence are analogous to before. Information states which imply fixed relative displacements are suitable for steady-state separation; a candidate could be momentum, but not position. If separation guidelines are violated there are no mechanisms to widen the gap.

This algorithm may be extended to guarantee that information states converge to a set difference, such that $(\xi_\mathrm{i} - \xi_j) \to \Delta_{ij}(t)$ as $t \to \infty$ [6].

$$\mathrm{u_i(t)} = \dot{\delta_\iota} - \sum_{j=1}^{n} a_{ij}\left[(\xi_\mathrm{i} - \xi_j) - (\delta_i - \delta_j)\right], \qquad i = 1, \dots, n$$

Here $\Delta_{ij} \triangleq \delta_i - \delta_j, \forall i \neq j$ and denotes setpoint state deviations. This algorithm is useful for maintaining relative vehicle positions, which has applications in formation control.

C. *Event-Triggered Consensus Protocols*

For resource conservation communications and corresponding actuator updates are spread over discrete time intervals, between which measured values are locked and held. There must be a sufficient frequency to effectively capture state changes and preserve performance goals. Conversely there must be a guaranteed lower bound on inter-event execution times to exclude Zeno behaviour, where infinite events happen on a finite time horizon [10].

The sampled-data approach is where transmissions are delayed by a fixed period, although improving utilisation this still results in an over-provisioning of resources. High frequency triggers required to maintain stability and provide new information in the transient phase are excessive as the system converges [10]. A dynamic approach is required to overcome this limitation.

An event-triggered protocol transmits information on demand to reduce wastage without jeopardising performance. This is characterised by an event-triggering condition which compares the error between the current and most recently transmitted information measurement (state or output) to a set threshold (e_T) [11].

$$error \geq threshold \triangleq \left|x(t) - \hat{x}(t_j)\right| \geq e_T$$

Here t_j describes the j^{th} consecutive sampling instant, $t \in [t_j, t_{j+1})$ for $j = 0,1, \ldots, \infty$. This provides a measure on how valuable the state of an agent is to maintaining overall closed loop behaviour and adapts tasks as required. A basic operating procedure is represented in Figure 1 below. The error threshold is selected to preserve underlying consensus stability concepts such as Lyapunov stability, input-to-state stability (ISS) and L_2 stability. The latter two are preferred in practical systems due to their robustness against external disturbances.

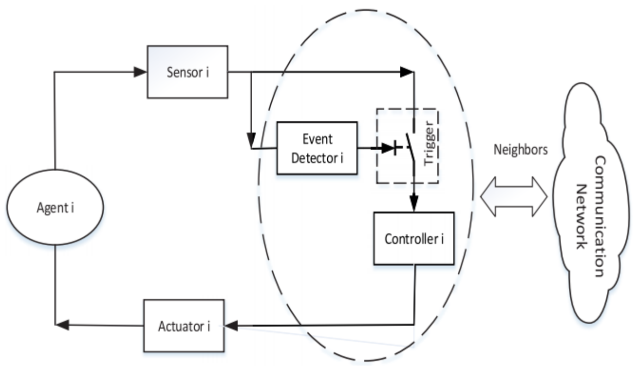

Figure 1 Event Triggering Process [10]

An early event triggered consensus strategy for a linear multi-agent system of N agents is considered in [11] for undirected, connected topologies. The solution is extensible to high order dynamics, described via

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \qquad i = 1,2, \ldots N$$

assuming

$$rank(AB) = rank(A)$$

It is also robust to switching topologies, and enforces a lower boundary on the inter-execution time to exclude Zeno behaviour for all agents. The event protocol defines a controller with latest broadcast values $\hat{x}$ and a triggering-threshold given as

$$\mathrm{u_i(t)} = -\mathrm{K}\sum_{j=1}^{n} a_{ij}(t)\left[\widehat{x_i}(t) - \widehat{x_j}(t)\right], \; i = 1, \ldots, n$$
$$e_T = k_i|z_i(t)|, i = 1,2, \ldots N$$

where

$$K \text{ is a gain matrix}$$
$$0 < k_i < \max\left(\lambda_N[L_p(t)]\right)$$
$$z_i(t) = \sum_{j=1}^{N} a_i(x_i(t) - x_j(t))$$

This relaxes the actuation update requirements, however $z_i(t)$ still requires constant knowledge of neighbouring states and global topology – contradicting motivations and leaving continuous communication as a requirement. Communication is relaxed in [12] to use only broadcasted states at the expense of a more complex triggering function. Assumptions of linearity and an undirected topology simplify event triggering, however, raise practical concerns.

An alternative strategy using a state-independent threshold for relative error is proposed in [13] for an undirected, connected topology. For single integrator agents the trigger protocol is given.

$$\mathrm{u_i(t)} = -\sum_{j=1}^{n}\left[\widehat{x_i}(t) - \widehat{x_j}(t)\right], \; i = 1, \ldots, n$$
$$e_T = (c_0 + c_1 e^{-\alpha t})$$

with constants

$$c_0 \geq 0, c_1 \geq 0, c_0 + c_1 > 0 \; and \; 0 < \alpha < \lambda_2(L_p)$$

By only needing local information to compute the state-triggering condition the need for continuous communication is eliminated. Design choices are required for c_0, c_1 and α to exclude the Zeno behaviour. If the choice for $c_0 \neq 0$ then consensus is not exact but bounded by a radius. Details are also given on extensions to accommodate for network stochasticity and double-integrator agents. The assumptions of linear dynamics and network topologies are again impractical.

D. Finite Time Convergence

The convergence rate is a significant performance metric for consensus control, so a typical problem is to develop controllers which drive the system in as little time as possible [6]. Most control schemes use Lipschitzian dynamics, leading to exponential convergence with infinite settling time [14] and dependence on initial state conditions [10]. With an event-triggering protocol which aims to reduce control updates, this may suffer further [10]. This motivates practical MAS design to reach consensus in finite time. Research in this area is attracting attention and evolving rapidly, with recent approaches in [15, 16, 17, 18].

An early event-triggered finite time consensus control scheme is proposed in [19] for a mobile sensor network. This designs a finite-time consensus algorithm and determines a state-dependent event-triggering rule to preserve stability. The triggering condition has increased complexity, and still requires continuous communication. Results show that when compared to a standard benchmark performance is maintained while reducing computational resource usage and control updates.

III. Research Problem

The aim of this project is to investigate a practical approach for a MAS to achieve consensus. As a foundational concept in unmanned traffic management (UTM) this is quintessential for the successful realisation of future UAM.

This research will simulate average consensus amongst homogeneous linear agents under a strongly connected network. Communication will be event triggered and guarantee convergence within finite time to achieve a balance of feasibility and performance. This will lay the foundation to explore a pressing design challenge which may come under one of the succeeding threads.

a) Robustness to disturbances
b) Robustness to network stochasticity
c) Relaxing network assumptions

The latter component is intentionally left open due to the rapid evolution of research in this realm.

IV. Investigation Program and Design

A. Vehicle Dynamics

The idealised rigid-body dynamics of a quadrotor vehicle [20] are a candidate to contextualise this project within UAM.

There are three degrees of freedom with cartesian x-y motion and rotation about a single axis. States are as follows.

$$\boldsymbol{z}' \triangleq [x \quad p_x \quad y \quad p_y \quad \theta \quad L]^T$$

These are x position, x momentum, y position, y momentum, pitch angle, and angular inertia. They have the following state derivates.

$$\dot{\boldsymbol{z}}' = \begin{bmatrix} \dot{x} \\ \dot{p_x} \\ \dot{y} \\ \dot{p_y} \\ \dot{\theta} \\ \dot{L} \end{bmatrix} = \begin{bmatrix} \dot{z_1} \\ \dot{z_2} \\ \dot{z_3} \\ \dot{z_4} \\ \dot{z_5} \\ \dot{z_6} \end{bmatrix} = \begin{bmatrix} m^{-1}z_2 \\ -(F_F + F_R)\sin(z_5) \\ m^{-1}z_4 \\ (F_F + F_R)\cos(z_5) - mg \\ J^{-1}z_6 \\ l_F F_F - l_R F_R - b_t J^{-1} z_6 \end{bmatrix}$$

Table 1. Quadrotor Variable Descriptions

Variable	Description
m	Vehicle mass
g	Gravity constant
J	Moment of inertia around centre mass
b_t	Aero-dynamic damping
F_F, F_R	Thrust force of front rotors, rear rotors
l_F, l_R	Distance from centre mass of front rotor, rear rotor thrust vectors

There is opportunity to simplify this model by assuming the pitch is a fixed value robust to disturbances. This could be analogous to platooning along an aerial highway. An implication is that $\dot{\theta} = \dot{L} = 0$, and so the front and rear motors must have an equal moment. This gives the resulting system.

$$\boldsymbol{z} \triangleq [x \quad p_x \quad y \quad p_y]^T$$

With changed state derivates this is a linear model. Momentums are single integrator dynamics, and positions are double integrator dynamics.

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \dot{x} \\ \dot{p_x} \\ \dot{y} \\ \dot{p_y} \end{bmatrix} = \begin{bmatrix} \dot{z_1} \\ \dot{z_2} \\ \dot{z_3} \\ \dot{z_4} \end{bmatrix} = \begin{bmatrix} m^{-1}z_2 \\ -2F sin(\bar{\theta}) \\ m^{-1}z_4 \\ 2F cos(\bar{\theta}) - mg \end{bmatrix}$$

The model is pictured below.

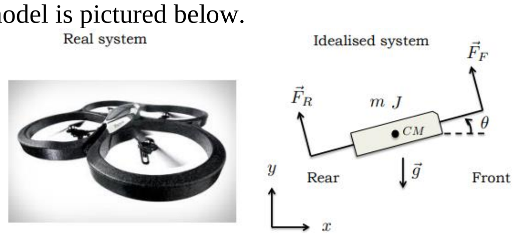

Figure 2 Quadrotor Model

B. Objectives, Methodology and Research Plan

The first objective of phase one is the setting up of a working environment to enable further building and testing. Tools used will be MATLAB and potentially Simulink. It will involve modelling and simulating of individual agents. Supplementary to this will be a method of visualising generic results and analysing them through a suite of automated unit tests (see following sub-section). To facilitate ongoing work care will be taken to observe code quality principles and make the environment adaptable, e.g. abstracting the dynamics of individual agents.

The remainder of phase one will focus on an initial event-triggered consensus implementation for linear, single integrator agents under a strongly connected time-invariant topology. Working towards this will first require the consensus algorithm detailed in II.B to be implemented. It will initially be continuous time, before moving to a discretised version. There are numerical examples in [6] which may be used to verify the approach. The state-independent triggering approach reviewed in II.C will be used to extend this. Again, there are numerical toy problems to emulate. Reproducing simple issues will be useful for establishing a foundation and asserting fundamental correctness for concepts.

Layers of complexity will be iteratively added and tested within project phase two to work towards the research goal. This will involve selecting a contemporary implementation of event triggered finite time consensus and analysing a gap in the work that corresponds to one of the research threads mentioned in section III. This is left general due to the project time scale and rapid rate at which new approaches are published. The literature review will be extended at the beginning of this phase to properly focus the investigation. Work updates will be saved to leave a trail of artefacts which may be used to benchmark performance and gain insights into advantages added.

A high-level process flow displaying tasks and interdependencies is pictured within Figure 3. This is broken down into a Gantt chart projected timeline, documented within appendix II.F.

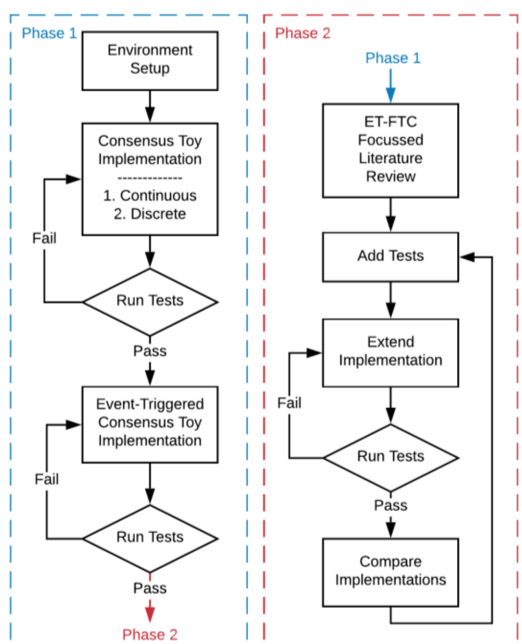

Figure 3 Project Workflow

Due to being an iterative process new learnings and insights will come as progress is made. This may cause the research question to evolve towards a more specific or interesting niche. In parallel there be regular updates to the literature review and math preliminaries to explore new strategies and develop comprehensiveness. With completed milestones new sections will be added to document work and results.

All relevant simulation materials will be stored within the GitHub repository https://github.com/ldale1/EGH400-Thesis-Project

C. Testing Suite

To verify the integrity of the implementation, a suite of unit tests will be developed using the inbuilt MATLAB testing framework.

Automated unit tests will be used to verify the solution correctness and coverage. These will check whether consensus is achieved using a range of total agents, initial state trajectories and network topologies. Convergence can be checked with the literature review definition, however with a sufficiently small error allowable for bounded consensus. With a limited set of numerical examples to base tests off, conclusions about novel test outcomes must be drawn from the principles reviewed.

It is impractical to verify the correct replication of a research paper via automated testing, as there is typically not exact data to compare results with. Important metrics such as the state or error over time are consistently displayed in plots rather than in a numerical form. As such, manual inspection will be required to draw conclusions.

The following table shows a preliminary list of tests. This will be expanded upon throughout the iteration process as complexity increases.

Table 2. Unit Tests

No.	Description	Pass Criteria
A	***Automated Unit Test Suite***	
A.1	*Agents*: 5 *Initial Values:* linear distribution *Network:* Undirected, connected	Consensus Reached
A.2	*Agents:*5 *Initial Values:* linear distribution *Network:* Undirected, disconnected	Consensus Not Reached
A.3	*Agents:* 5 *Initial Values:* exponential distribution *Network:* Undirected, connected	Consensus Reached
A.4	*Agents:* 10 *Initial Values:* linear distribution *Network:* Undirected, connected	Consensus Reached
B	***Manual Unit Test Suite***	
B.1	State trajectory plot	Mirrored Plot
B.2	Transmission triggering plot	Mirrored Plot
B.3	State error plot	Mirrored Plot

V. Conclusion

This proposal presents a methodology to investigate event-triggered consensus which guarantees finite time convergence. This will be purely research based, with objectives related to efficient and practical realisation of UAM. The project will run over two phases and involves developing an initial implementation which will be iteratively built upon.

APPENDIX I[3]

A. Graph Preliminaries

A team of p vehicles may have information exchanges modelled by graphs (V_p, E_p) which are either directed or undirected. Both cases have a node set $V_p = \{1, \dots, p\}$ and edge set $E_p \subseteq V_p \times V_p$. In the directed case edge (i, j) denotes that child node j may receive information from parent node i, although not vice versa. In the undirected case edge (i, j) signifies both nodes may receive information from one another. This may be categorised as a special directed graph case, where undirected (i, j) implies a directed couple (i, j) and (j, i). Self-edges are not allowed, unless stated. A weighted graph maps a weight to every edge.

Directed and undirected paths are sequences of edges (i_1, i_2), (i_2, i_3), … in directed and undirected graphs respectively. These are cycles instead if starting and ending at the same node.

A directed tree is where every node has a parent except for a root node, which consequentially has s directed path to its all its descendants. A directed graph is strongly connected if every node is the root of a directed tree reaching all other nodes. An undirected tree is where every pair of nodes is connected by a path. This is analogous to an undirected graph being connected.

A. Matrix Preliminaries

The adjacency matrix $\mathrm{A}_p = \left[\alpha_{ij}\right] \in \mathbb{R}^{p \times p}$of a graph reflects the weights of edges (j, i). Value a_{ij} is 0 if $(j, i)\notin E_p$, as self-edges are not allowed this causes the diagonal to be zeros. For the undirected graph the adjacency matrix is symmetrical as $(j, i)\in E_p$ requires $(i, j)\in E_p$ causing $a_{ij} = a_{ji}$. They are also balanced, meaning $\sum_{j=1}^{p} a_{ij} = \sum_{j=1}^{p} a_{ji}$ for all i.

There is a corresponding Laplacian matrix $L_p = \left[l_{ij}\right] \in \mathbb{R}^{p \times p}$ which is defined as $L_p \triangleq D - \mathrm{A}_p$. Here the in-degree matrix $D = \left[d_{ij}\right] \in \mathbb{R}^{p \times p}$ is given as $d_{ij} = 0, i \neq j$ and $d_{ii} = \sum_{j=1}^{p} a_{ij}\,, i = 1, \dots, p$. This is not the common Laplacian matrix definition, however it has relevance to consensus algorithms.

A row stochastic matrix $D = [d_{ij}] \in \mathbb{R}^{p \times p}$ is nonnegative and has every row sum to one. The product of these are still row stochastic. It is indecomposable and aperiodic (SIA) if $\lim_{k\to\infty} D^k = \mathbf{1}y^T$ for $y \in \mathbb{R}^n$. They have the eigenvalue 1 for eigenvector $\mathbf{1}_n$. This matrix is used to study consensus in settings where the topology evolves and is subject to disturbances.

[3] The work in this appendix is adapted exclusively from [6].

APPENDIX II

A. Progress Summary

This project has progressed to the stage where work on the implementation is possible. The foundation for this was established within Quarter 1.

B. Quarter 1 (Semester 2, 2020 - Project Proposal)

This quarter completed the initial engineering literature review worksheet and established a project proposal. A clear plan was established for completion of the investigation.

C. Quarter 2 (Semester 2, 2020 - Progress Report)

N/A

D. Quarter 3 (Semester 1, 2021 - Progress Report)

N/A

E. Quarter 4 (Semester 1, 2021 – Final Report)

N/A

F. *Project Timeline*

Student | Phase 1

	Task name	Start date	End date	Duration
		07/09/2020	**02/11/2020**	**56d**
1	**Phase 1: Quarter 2**	**07/09/2020**	**02/11/2020**	**56d**
1.1	2A Oral Report	07/09/2020	29/10/2020	52d 23h
1.2	2B Written Report	07/09/2020	01/11/2020	55d 23h
1.3	**Simulation Setup**	**07/09/2020**	**17/09/2020**	**10d**
1.3.1	Scripting Setup	07/09/2020	10/09/2020	3d
1.3.2	Write Automated Tests	10/09/2020	17/09/2020	7d
1.3.3	Write Manual Tests	10/09/2020	17/09/2020	7d
1.4	**Consensus Implementation**	**17/09/2020**	**04/10/2020**	**17d**
1.4.1	Continuous Time	17/09/2020	24/09/2020	7d
1.4.2	Discrete Time	24/09/2020	29/09/2020	5d
1.4.3	Testing & Refinement	29/09/2020	04/10/2020	5d
1.4.4	Document Work	29/09/2020	04/10/2020	5d
1.5	**Event-Triggered Implementation**	**04/10/2020**	**02/11/2020**	**29d**
1.5.1	Controller	04/10/2020	11/10/2020	7d
1.5.2	Fixed Trigger	04/10/2020	11/10/2020	7d
1.5.3	Event Trigger Design	11/10/2020	19/10/2020	8d
1.5.4	Async Event Triggers	15/10/2020	23/10/2020	8d
1.5.5	Testing and Refinement	23/10/2020	28/10/2020	5d
1.5.6	Document Work	23/10/2020	02/11/2020	10d

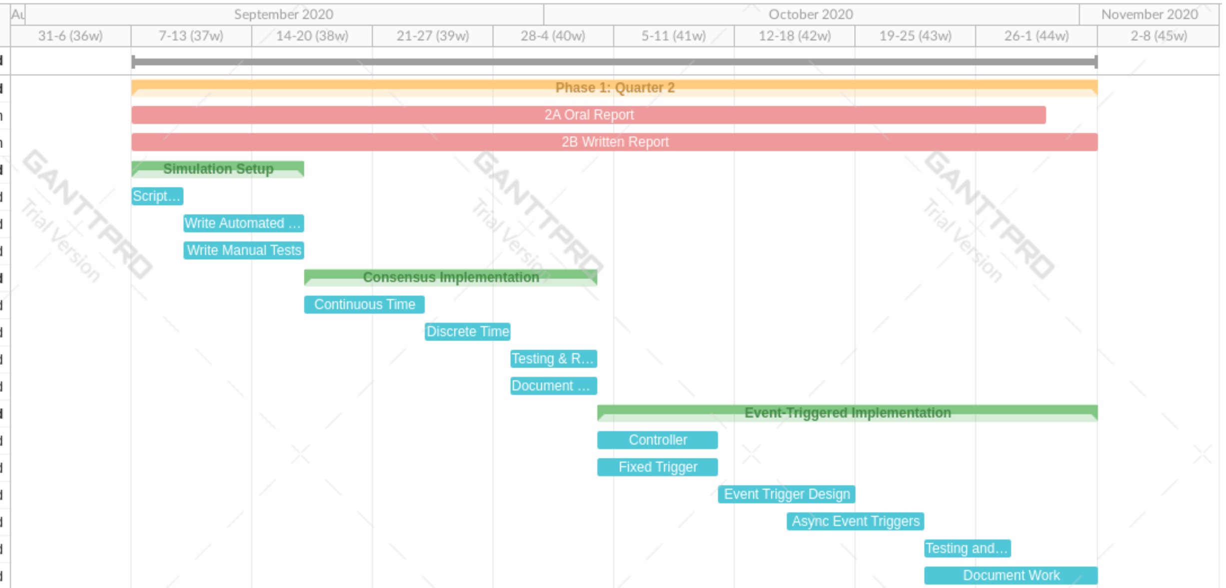

Figure 4 Gantt Chart Remainder Phase 1

Student | Phase 2 (Projected)

	Task name	Start date	End date	Duration
		01/03/2021	**04/06/2021**	**13w 4d**
1	**Phase 2: Quarter 3**	**01/03/2021**	**12/04/2021**	**6w**
1.1	**Task 1**	**01/03/2021**	**16/03/2021**	**2w 1d**
1.1.1	General Reading	01/03/2021	08/03/2021	1w
1.1.2	Source Identification	05/03/2021	11/03/2021	6d
1.1.3	Gap Identification	10/03/2021	15/03/2021	5d
1.1.4	Documentation Upd…	11/03/2021	16/03/2021	5d
1.2	**Finite Time ETC Impl…**	**16/03/2021**	**12/04/2021**	**3w 6d**
1.2.1	Design	16/03/2021	25/03/2021	1w 2d
1.2.2	Implementation	20/03/2021	08/04/2021	2w 5d
1.2.3	Testing and Refine…	04/04/2021	12/04/2021	1w 1d
1.2.4	Documentation	04/04/2021	12/04/2021	1w 1d
2	**Phase 2: Quarter 4**	**12/04/2021**	**04/06/2021**	**7w 4d**
2.1	**Design Robustness**	**12/04/2021**	**04/06/2021**	**7w 4d**
2.1.1	Survey Solutions	12/04/2021	21/04/2021	1w 2d
2.1.2	Implement, Iterate	21/04/2021	26/05/2021	5w
2.1.3	Testing and Refine…	24/04/2021	04/06/2021	5w 6d
2.1.4	Document	17/05/2021	04/06/2021	2w 4d

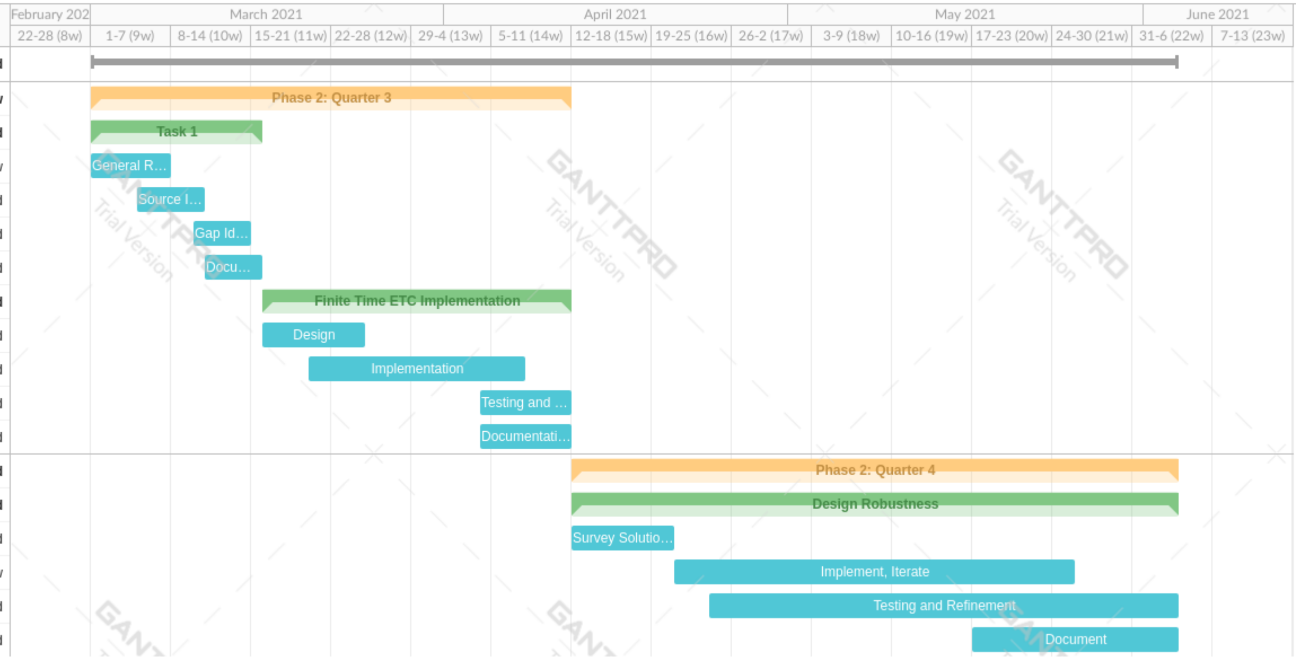

Figure 5 Gantt Chart Projected Phase 2

References

[1] A. Straubinger, R. Rothfeld, M. Shamiyeh, K.-D. Buechter, J. Kaiser and K. O. Ploetner, “An overview of current research and developments in urban air mobility – Setting the scene for UAM introduction,” *Journal of Air Transport Management,* vol. 87, 2020.

[2] V. Bulusu, “Urban Air Mobility: Deconstructing the Next Revolution in Urban Transportation - Feasibility, Capacity and Productivity,” ProQuest LLC, Ann Arbor, 2019.

[3] Uber Elevate, "Fast forwarding to the future of on-demand, urban aviation," Uber Technologies Inc, 2020. [Online]. Available: https://www.uber.com/in/en/elevate/uberair/. [Accessed 30 August 2020].

[4] Airbus, “Urban Air Mobility – the sky is yours,” Airbus Group, 27 November 2018. [Online]. Available: https://www.airbus.com/newsroom/stories/urban-air-mobility-the-sky-is-yours.html. [Accessed 30 August 2020].

[5] X. Yang and P. Wei, “Scalable Multi-Agent Computational Guidance with Separation Assurance for Autonomous Urban Air Mobility,” *American Institute of Aeronautics and Astronautics,* vol. 43, no. 8, pp. 1473-1486, 2020.

[6] W. Ren and R. W. Beard, Distributed Consensus in Multi-vehicle Cooperative Control, London: Springer, 2008.

[7] A. Bemporad, M. Heemels and M. Johansson, Networked Control Systems, Berlin: Springer, 2010.

[8] X. Wang and M. D. Lemmon, “Event-Triggering in Distributed Networked Control Systems,” *IEEE Transactions on Automatic Control,* vol. 56, no. 3, pp. 586-601, 2011.

[9] L. Ding, Q.-L. Han, X. Ge and X.-M. Zhang, “An Overview of Recent Advances in Event-Triggered Consensus of Multiagent Systems,” *IEEE Transactions on Cybernetics,* vol. 48, no. 4, pp. 1110-1123, 2018.

[10] X. Ge, Q.-L. Han, D. Ding, X.-M. Zhang and B. Ning, “A survey on recent advances in distributed sampled-data cooperative,” *Neurocomputing,* vol. 275, pp. 1684-1701, 2018.

[11] Z. Zhang, F. Hao, L. Zhang and L. Wang, “Consensus of linear multi-agent systems via eventtriggered control,” *International Journal of Contro,* vol. 87, no. 6, pp. 1243-1251, 2014.

[12] S. S. Kia, J. Cortes and S. Martinez, “Distributed event-triggered communication for dynamic average,” *Automatica,* vol. 59, pp. 112-119, 2015.

[13] G. S. Seyboth, D. V. Dimarogonas and K. H. Johansson, “Event-based broadcasting for multi-agent average consensus,” *Automatica,* vol. 49, no. I, pp. 245-252, 2012.

[14] V. T. Haimo, “Finite Time Controllers,” *SIAM Journal on Control and Optimization,* vol. 24, no. 4, pp. 760-770, 1986.

[15] J. Lui, Y. Yu, H. He and C. Sun, “Team-Triggered Practical Fixed-Time Consensus of Double-Integrator Agents With Uncertain Disturbance,” *IEEE Transactions on Cybernetics,* vol. Early Access, pp. 1-10, 2020.

[16] X. Lu, “Distributed Event-Triggered Control for Prescribed Finite-Time Consensus of Linear Multi-Agent Systems,” *IEEE Access,* vol. 8, pp. 129146 - 129152, 2020.

[17] C. Du, X. Liu, W. Ren, P. Lu and H. Liu, “Finite-Time Consensus for Linear Multiagent Systems via Event-Triggered Strategy Without Continuous Communication,” *IEEE Transactions on Control of Network Systems,* vol. 7, no. 1, pp. 19-29, 2020.

[18] J. Liu, Y. Zhang, Y. Yu and C. Sun, “Fixed-Time Event-Triggered Consensus for Nonlinear Multiagent Systems Without Continuous Communications,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 11, pp. 2221-2229, 2018.

[19] Q. Lu, Q.-L. Han, B. Zhang, D. Liu and S. Liu, “Cooperative Control of Mobile Sensor Networks for Environmental Monitoring: An Event-Triggered Finite-Time Control Scheme,” *IEEE Transactions on Cybernetics,* vol. 47, no. 12, pp. 4134-4147, 2017.

[20] T. Perez, “Modelling of Physical Dynamical Systems,” Queensland University of Technology, Brisbane, 2017.