1-2021

Feasibility Study of Serial Hybrid-Electric Systems in Small Aircraft

Kyle Rosenow

SANTA CLARA UNIVERSITY

Department of Mechanical Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Kyle Rosenow

ENTITLED

FEASIBILITY STUDY OF SERIAL HYBRID-ELECTRIC SYSTEMS IN SMALL AIRCRAFT

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMETNS
FOR THE DEGREE OF

MASTERS OF SCIENCE
IN
MECHANICAL ENGINEERING

	12-Jan-2021
Thesis Advisor: Godfrey Mungal	date
	1/18/2021
Thesis Reader: ~~Nik~~ Djordjevic	date
	2/1/2021
Department Chair: Drazen Fabris	date

Abstract

Three different small aircraft, a Diamond DA40, a Cessna Skyhawk 172S and a Cirrus SR22, are used to assess the feasibility of converting an existing aircraft's power system to a serial-hybrid system or an all-electric system. The serial-hybrid system uses a gasoline engine to generate electricity that can power the main electric motor or charge onboard batteries, while the all-electric system uses batteries only and does not carry a gasoline engine. General system designs are proposed, and a calculation model was developed to allow for analysis of the three different aircraft and their variants. The all-electric and serial-hybrid variants are compared to the existing aircraft, the gas variant, by replicating the gas variant's performance on a representative flight plan as best as possible. Feasibility is evaluated on how well the variants perform relative to the gas variant and how power plant system weight, useable weight, endurance, range, and fuel consumption compare. Converting to an all-electric would reduce an aircraft's basic empty weight, but battery packs require large amounts of weight to achieve similar amounts of flight time. A serial-hybrid possesses a higher basic empty weight but will be able to trade battery pack weight for gasoline weight, and as a result can receive some benefits of an all-electric and benefits of an all-gas system. Performing a conversion of a gas system to an all-electric system would be difficult to achieve successfully without sacrificing significant performance such as speed and flight endurance. However, a serial-hybrid system conversion is possible, but flight endurance and range are sacrificed while fuel consumption is reduced. A serial-hybrid is useful in some scenarios, such as a training aircraft, due to low time per flight and short distances of flight, but a gasoline powered aircraft can travel farther and for longer due to the higher energy density of gasoline.

Acknowledgements

Thank you to Santa Clara University Professor Tim Healy for providing early technical guidance on electric vehicle battery systems and to Lockheed Martin Fellow and FAA Certified Flight Instructor Dr. Larry Capots for technical guidance on aircraft systems analysis. Thank you as well to this report's reader Nik Djordjevic and thesis advisor, Professor Godfrey Mungal.

Table of Contents

List of Figures

List of Tables

Variables and Acronyms

C_{D_I} = Coefficient of Drag, Induced

C_{D_o} = Coefficient of Drag

C_{D_p} = Coefficient of Drag, Parasite

C_{ℓ} = Coefficient of lift, 2D airfoil

C_L = Coefficient of Lift, 3D airfoil

S_f = Suction Force

S_{ref} = Wing Reference Surface Area

S_{wet} = Wetted Surface Area

V_A = Maneuvering speed

V_S = Stall speed

V_{cruise} = Cruise Speed

V_v = Vertical Velocity

V_x = Best Climb Speed for Shortest Horizontal Distance Travel

V_y = Best Climb Speed for Quickest Time-To-Climb

W_{PP} = Power Plant Weight

W_{batt} = Battery Weight

p_o = Pressure at Sea-Level

ρ_{LS} = Light-Sport Engine Power Loading Density

ρ_{batt} = Battery Power Density

ρ_o = Air Density at Sea-Level

a = Acceleration

A = Aspect Ratio

A = Disk Area

AC = Alternating Current

Ah = Amp-Hours

BEW = Basic Empty Weight

BMS = Battery Management System

CAS = Calibrated Airspeed

DC = Direct Current

e-fan = Electric Fan

eVTOL = Electric Vertical Takeoff and Landing

g = Acceleration Due to Gravity

GA = General Aviation

h = Elevation Change

IAS = Indicated Airspeed

LiPo = Lithium Polymer

$\dot{m}$ = Mass Flow Rate

MGW = Maximum Gross Weight

POH = Pilot Operating Handbook

t = Motor Run Time

TAS = True Airspeed

TBO = Time Between Overhaul

F = Force or Thrust

K = K-factor

L = Lift

M = Mach number

P = Power

Q = Air Flow Rate

S = Surface Area

SoS = Speed of Sound

T = Thrust

V = Velocity

W = Weight

p = Pressure at Altitude

α = Angle of Attack

γ = Pitch Angle

ρ = Air Density at Altitude

η = Efficiency

US-EPA = United States Environmental Protection Agency

FAA = Federal Aviation Administration.

1 Introduction

The aviation industry in the United States is facing increased environmental regulations with other regulators around the world already instituting stricter emission standards [1]. A possible path to meet these standards is the electrification of aircraft. Such technology is currently used in automobiles, and while mostly prototypes exist for aircraft. Electric aircraft are more sensitive to the drawbacks of electric propulsion than electric automobiles. Electric propulsion systems are heavy, and weight is a major factor in aircraft design, operation, and efficiency. This report will explore a one-to-one exchange in an existing aircraft from an internal combustion engine to an electric motor system or hybrid-electric system and analyze the trade-offs. While the cited emission standard by the US-EPA is targeting large airliner-type aircraft, general aviation will likely see regulations in the future for newly produced aircraft.

The aircraft examined in this report are small aircraft belonging to the segment of general aviation that can be flown by a single pilot, with a private pilot certificate, can carry three additional people, is on the order of 1200 kg (2600 lbs) max gross weight, and stays below 460 km/h (250 kts) during cruise. These aircraft are typically what you would see at a small airport that are used for personal activities and transportation or are used for flight training.

The proposed hybrid system would utilize battery storage and an onboard generator to convert fuel energy into electric energy thereby achieving a higher fuel efficiency compared to a gas engine while sacrificing less range and useful load compared to an all-electric aircraft. As with road vehicles, a hybrid system could provide a useful intermediate step to 100% aircraft electrification while battery technology improves, and benefits from the advantages of hydrocarbon-fuel engines. As it stands right now, energy storage using a battery is lower in energy-capacity per unit-of-weight and per unit volume than a similar weight or a similar volume of a hydrocarbon fuel. By using three existing aircraft as baselines, a hypothetical system conversion is performed to show trade-offs in using an all-electric system, a serial-hybrid system, and the existing internal combustion engine system.

Hybrid systems in general fit into two broad classes: parallel hybrids and serial hybrids. A parallel hybrid aircraft is where a fuel engine and electric motor both directly provide power to the propeller. In a car, the gas or diesel engine and the electric motor directly drive the wheels. The Toyota Prius [2] is an example of parallel hybrid system in a car where the electric motor is used for low speeds and starting the car's movement and then the gas engine takes over at higher speeds and during sustained driving. A serial-hybrid aircraft is where the electric motor only directly powers the propeller, and the fossil-fuel engine generates electricity for use by the electrical motor. The Chevy Volt [3] is an example of a serial-hybrid car since the car contains a "range extender" engine that can provide electricity to the drive system when the onboard batteries are depleted enabling a longer driving distance. A serial hybrid system is the focus for this report.

To address the question of feasibility, the proposed systems can be evaluated based on several factors: overall weight, useable weight, range, endurance, and economics. Overall weight is the weight of the power plant system, and includes the power plant itself (electric motor, or gas engine), the energy storage (batteries and fuel), and the other parts needed to make the system function (such as motor controller, generator, and piping.) However, the pilot of the aircraft usually cares about the usable weight, which is the weight available for people, cargo, and fuel (if fuel is used). The range is the distance the aircraft can fly, and the endurance is the time the aircraft can fly. If someone were to trade in their existing GA aircraft for an all-electric or hybrid-electric aircraft, they will want to know if it can suit their needs for travel. Therefore, feasibility will be evaluated using the stated parameters and comparing the serial-hybrid and all-electric system to the gas version of the existing aircraft. The airframe and general layout of the aircraft will stay the same and a custom, new aircraft design is not proposed.

Additionally, the appeal of electric aircraft is that the projected cost-per-hour of operation is lower because the overall system is less complex and maintenance costs are lower [4,5]. Explicit operational costs beyond fuel consumption is not considered in the report as these costs are highly variable, and Section 1.1 outlines the source of this uncertainly in a brief discussion on overall aircraft ownership. Overall, environmental stewardship, complying with possible future regulations around the world, and reduced operating costs are the main reasons to consider GA aircraft electrification.

1.1 Operations and Maintenance Costs

Estimating operational and maintenance costs for general aviation aircraft is difficult due factors such as regional fuel prices, regional maintenance labor costs, the complexity of an aircraft, certification status of the aircraft, insurance, and any financing costs. Even within the same family of aircraft, built in the same year, differences in avionics and other addon features influence operational costs between otherwise identical aircraft.

Mandated by the FAA, maintenance such as annual inspections, pitot-static system inspections, and emergency locator inspections must be performed as specific calendar intervals [6,7]. Other inspections, like the time between overhauls (TBO), are dependent on the frequency of flying, and is an interval recommended by the manufacturer stating that an aircraft's engine should be disassembled, inspected, repaired, and rebuilt. Based on anecdotal evidence by talking to aircraft owners the author knows personally, the cost of an engine overhaul seems to increase as engine power or complexity increases. The information that follows in this section regarding operation costs is provided as contextual information and not based on rigorously determined data.

The cost of an overhaul for an aircraft engine on the order of 75 kW (100 HP) is around $30,000 to $40,000 and an aircraft with an engine of 134kW (180 HP) is approximately $40,000 to $60,000. These costs are driven by labor and the parts needed to disassemble and reassemble aircraft engines. By contrast, an electric motor system consists of batteries, wiring, solid-state control circuitry and the electric motor. The rotating rotor inside the electric motor is the main moving part compared to the intricate internal combustion engine with many moving parts. Time between inspections and overhauls (except for regulation imposed inspections) is less, meaning less recurring cost.

The unique source of recurring cost for electrified aircraft will be that battery packs need replacement since charge-discharge cycles reduce battery capacity. At the time of this writing, Tesla, Inc. is providing a 150,000-mile warranty on their Model S electric vehicle that the battery will retain 70% capacity [8]. The Model S advertised range is 400 miles [9]. Therefore, Tesla, Inc. is guaranteeing a minimum of 375 charge-discharge cycles assuming capacity remains at the maximum. While this is a warranty and not the actual lifetime of a battery, it indicates how much confidence Tesla, Inc. has in its batteries. A research paper by Harlow et al. [10] is showing a comparison between two cell configurations and typical cylindrical cells lose 50% capacity at 1500 cycles and a pouch configuration lose only 10-15% capacity at 4000-4500 cycles. Increasing charge-discharge cycles will directly reduce the cost of the battery pack over the lifetime of the vehicle and the cost of replacement. Cost of ownership in this report will focus only on fuel savings, but battery technology is evolving and will increase the appeal and feasibility of aircraft electrification.

1.2 Current Technology and Active Development Areas

Electrification of aircraft propulsion is an active area of research and commercialization of new technology. New businesses are starting in different areas of the electric aircraft market as well as investment from prominent aircraft companies. While hybrid aircraft is the focus of this report, technology being developed for electric aircraft influences the feasibility of hybrid systems.

1.2.1 Vertical Take-off and Landing

At the time of this writing, electric vertical-takeoff-and-landing (eVTOL) is an evolving field with many businesses working on concepts and protypes for this type of aircraft. Kitty Hawk [11] and Joby Aviation [12] are two such companies that are working on these concepts, and their goals are short distance transport using all-electric aircraft. By contrast, the focus of this report is on existing fixed wing, horizonal take off and landing, single-engine aircraft.

1.2.2 In-Development Passenger-Service Aircraft

Some of the new concepts for passenger aircraft designed around hybrid and all-electric systems are relevant to potential future designs of GA aircraft. The passenger aircraft described next are serial-hybrids, which means they will be generating electricity using an electrical generator, such as an Auxiliary Power Unit (APU), to provide power to batteries and the electric motors. The hydrocarbon engine will not directly operate the propulsion systems.

Wright Electric [13] is developing a distributed electric fan (e-fan) propulsion system for a 186-seat passenger jet which lacks the usual vertical tail that is present on current passenger aircraft. A distributed propulsion system uses an increasing number of smaller propulsion devices instead of 2 to 4 large engines and as a result the vertical tail is less necessary since if one, smaller engine stops working, the aircraft is less affected by the unequal amounts of thrust on each side of the aircraft. A smaller or different shaped vertical tail reduces drag allowing the aircraft to fly further or faster. The vertical fin is however still required to orient the aircraft in the direction of travel much like a wind vane orients in the direction of the wind. A distributed engine system design contrasts to the more traditional design by Zunum Aero [14] and Airbus [15] that use the normal tail design with 2 to 4 e-fan engines. Zunum's aircraft seats 9 people and the Airbus E-FanX is projected to carry 186 passengers.

The turbofan engine powers many aircraft today and works by a jet engine spinning a large multi-bladed propeller inside a shroud. The electric fan concept replaces the jet engine with an electric motor enabling a compact design and does not require air to operate like a jet engine requires air for combustion. The benefit of this design is that propulsion motors and air-inlets can decouple meaning an air-inlet is not needed for each engine, allowing reduced air-inlet drag [4,16].

1.2.3 Existing or In-Development General Aviation-type Aircraft

Bye Aerospace [17] is working on a 2-seat and a 4-seat electric aircraft that is predicted to have a 3-hour (cruise) flight time. Bye is accepting orders for their two and four-seater aircraft with anticipation of delivering the first two-seater aircraft in 2021. Their promotional material shows a working prototype of the two-seater aircraft.

Eviation [18] is working on a 9-seater, three engine electric aircraft, and they are working on a prototype to conduct their first test flights. This is an all-electric aircraft with the unique feature of putting one electric motor on each wing tip due to the small weight and size of electric motors. Their claim is that this reduces induced drag from wing tip vortices and that the motors can help with yaw control[5,18].

AMPAIRE [19] is developing and testing a parallel-hybrid aircraft based on the Cessna 337 Skymaster that looks to use the standard gas engine on the front and a second electric motor on the back. It is a dual engine design where one propeller is in front of the engine “pulling” the aircraft along and the other propeller is behind the other engine “pushing” the airplane. Voltaero [20] is developing a parallel-hybrid that appears similar to AMPAIRE’s aircraft, but they are using three electric motors - a pull-prop electric motor on each wing, and a push-prop gas engine behind the cabin.

Rolls-Royce [21] is working on a high-performance single seat electric aircraft called Accel that will be capable of higher speeds and aerobatics. This appears to be a project to demonstrate the technology and to experiment with new technology. Siemens was previously working on a project like the Accel aircraft, but Rolls-Royce purchased Siemens e-aircraft division in 2019.

Currently flying in the United States and Europe is the Pipistrel Alpha Electro aircraft [22]. It is a light sport, all-electric aircraft aimed at the pilot training market. Just large enough to fit two people, the plane can fly for 1 hour with 20 minutes reserve doing traffic pattern practice, or 45 minutes plus 20 minutes reserve cruising.

Currently under development by Pipistrel is a newer plane called Panthera. Right now it’s a gas only aircraft, but there are plans and figures available for a series-hybrid and all-electric variant of the aircraft [23,24]. Table 1-1 is a comparison taken directly from Panthera’s website showing the stats of each aircraft variant. These numbers can be referenced later to compare results, and it shows a decrease in performance when compared to the working gas variant.

Table 1-1: Panthera Performance			
	Panthera	**Panthera Hybrid**	**Panthera Electro**
Category	Utility (+4.4 g.)	Utility (+4.4 g)	Utility (+4.4 g)
Power plant	Lycoming IO-540	Hybrid 145 kW	Pure electric 145 kW
Rated power	210 HP	195 HP (equivalent)	195 HP (equivalent)
Specifications			
Max Take-off Weight	1200 kg / 2640 lb	1200 kg / 2640 lb	1200 kg / 2640 lb
Useful payload	520 kg / 1145 lb	270 kg / 595 lb	200 kg / 440 lb
Full fuel payload	345 kg / 760 lb	n/a	n/a
Performance (Max Take-off Weight)			
Typical cruise speed (TAS)	374 km/h / 202 kts	263 km/h / 142 kts	218 km/h / 118 kts
Climb rate at MTOW	6.1 m/s / 1200 fpm	5.7 m/s / 1140 fpm	5.7 m/s / 1140 fpm
Range at cruise speed, 4 people aboard (incl. 45 min reserve)	>1900 km / >1025 NM	1220 km / 660 NM	400 km / 215 NM
Service ceiling	6,100 m/ FL 200	4000 m / FL 130	4000 m / FL 130

2 Aircraft Models, Configurations, Design Goals, Components

Assessing the feasibility of a serial-hybrid system on existing aircraft is the goal of this study, and to perform the analysis, three different aircraft are discussed and will be compared by performing a representative mission for a single engine, general aviation (GA) aircraft. An additional configuration, the all-electric system, is considered as a comparison since all-electric aircraft already exist or are in development, and an electric variant is closely related to the serial hybrid variant discussed in this report.

2.1 General Aviation Flight Profiles

Small general aviation aircraft serve three broad purposes: Recreation, personal transportation, and flight training. For recreation and flight training, these flights typically stay around 1200 meters or less in altitude and remain close to the home airport. Flight time, or endurance, best describes the capability of these flights and is simply how long the aircraft can maintain powered flight.

A common flight for short recreation and flight training activities is flying in an airport's traffic pattern. The traffic pattern is a methodical way for aircraft to fly near an airport and to land. An aircraft will take-off, climb to 300 meters above ground level, fly parallel to the runway, opposite the direction of take-off, start descending, and land again on the runway. Figure 2-1 depicts this flight path from a top-down view.

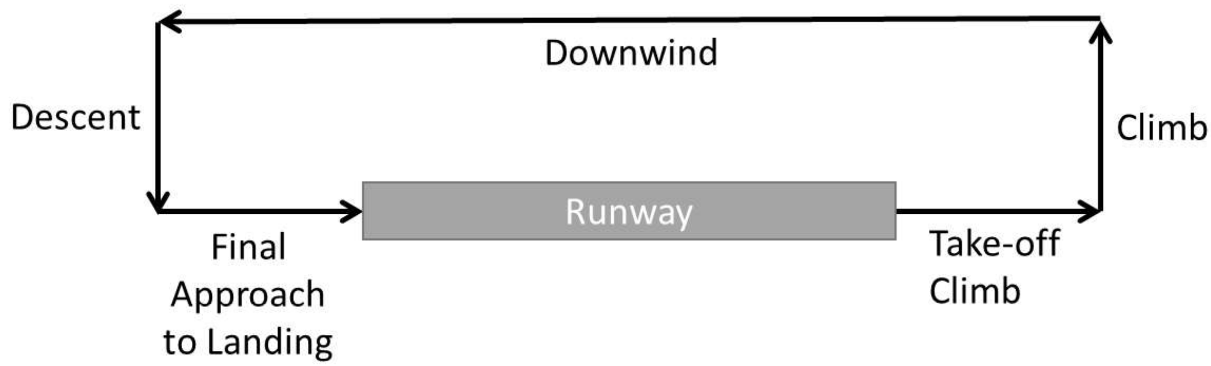

Figure 2-1: Schematic of a traffic pattern flight path. The downwind segment is at the traffic pattern altitude, and in this report can be treated as the cruise altitude.

Personal transportation is more concerned with the distance that can be traveled and the speed at which the aircraft can travel. The aircraft's range is how far it can travel and is affected by the wind speed in flight. The speed reported inside the aircraft is the speed relative to the outside air immediately around the aircraft. The ground speed is the actual speed of the aircraft. (For the purposes of this report, the ground speed and the airspeed are assumed equal.) The range then can be estimated as the airspeed multiplied by the time of flight.

The flight profile illustrated in Figure 2-2 will be the profile used primarily for modeling and addressing the question of feasibility of a hybrid system. The segments of the flight are the

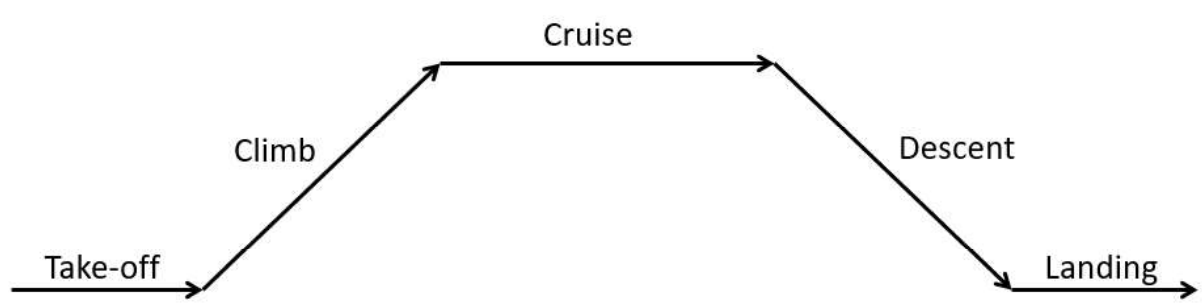

Figure 2-2: The major components of a GA aircraft's flight path from left to right.

same regardless if the airplane is staying local, or traveling from point A to point B, with the main difference being the altitude reached during the cruise segment. Figure 2-2 consists of five segments which are takeoff, climb, cruise, descent, and landing. In the first phase, the aircraft will take-off, and transition into the second phase to maintain a steady climb that is based on the maximum power output of the aircraft's propulsion system. The third phase is cruise where the aircraft will maintain an altitude of 2500 meters (8000 feet) at an airspeed faster than the climb phase while utilizing approximately 65% to 85% of maximum power. The fourth phase is descent where the aircraft will descend to the airport at a specified airspeed and vertical velocity. The fifth phase is when the aircraft lands at the airport.

The serial-hybrid and all-electric aircraft will be compared to the existing aircraft (referred to as the "gas variant" in this report) by switching the existing power plant with an equivalent-in-power all-electric or hybrid-electric power plant. The power plant is connected to the same propeller among all variants.

Take-off, the first phase, will not be considered in significant detail as the primary parameters affecting this phase are thrust, rolling friction, and aerodynamic drag. Thrust is influenced by the power available in the power plant and the type of propeller, but since the power output of the power plant and the propeller are the same between variants, thrust will not affect take-off performance. The next parameter is rolling friction which is a function of the maximum gross weight (MGW) of the aircraft. The max gross weight between variants will be the same since this study alters the existing aircraft as little as possible. In addition, the max gross weight is the worst-case scenario at takeoff and anything lighter will perform better than conditions at max gross weight. Lastly, aerodynamic drag is a function of the aircraft's shape and the aircraft shape is not being altered. Overall, takeoff performance will be the same among an individual aircraft's variants.

Climb, cruise, and descent are discussed in detail in Sections 3.2.3, 3.2.4, and 3.2.5 respectively in relation to the equations and model outlined in Section 3. These phases are where the large majority of energy is used during a flight. The fifth phase, landing, will not be considered in detail for similar reasons as take-off. Landing is assumed to be at max gross weight and would be the worst-case scenario for a landing. The primary goal of this study is to determine the feasibility of a hybrid-electric power system in-flight.

2.2 Configurations

Three configurations are considered to address both the accuracy of the model and to answer the question of feasibility: gas-powered, serial hybrid and all-electric. The gas-powered configuration can verify the accuracy of the model's predictions when compared to the performance of existing aircraft and thus acts as the baseline configuration. The serial-hybrid variant functions by an electric motor directly spinning the propeller while the gas engine is an electricity generator. The all-electric operates with only batteries as the exclusive power source for the main electric motor.

2.2.1 Gas Variant

The gas engine power plant, represented by a block diagram in Figure 2-3, shows a how a gas

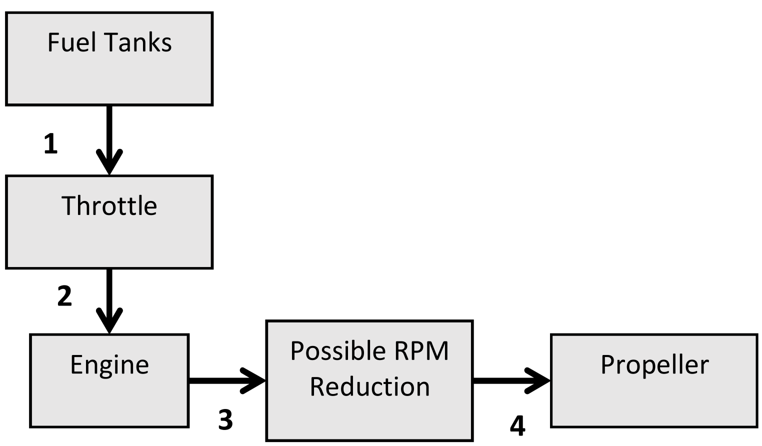

Figure 2-3: System diagram of gas engine. The numbers correlate to the effeciencies listed in Table 2-1.

aircraft is represented in the model and illustrates how the efficiencies of energy (power) transfer through the system. The energy source (fuel) moves through the system where the engine converts the chemical energy into mechanical energy to operate the propeller. The direction of energy flow is represented by the arrows and are labeled by numbers, which correspond to the efficiencies listed in Table 2-1. Arrow 2 represents the engine efficiency of converting supplied energy (fuel) into useful energy for the next block. The amount of mechanical energy converted from chemical energy via fuel combustion is 30% or 0.3 and is a general efficiency for combustion engines.

Table 2-1: Efficiencies for gas system	
Label	Efficiency
1	n/a
2	0.3
3 & 4	0.8

Arrows 3 and 4 are grouped together in Table 2-1 and assigned an overall efficiency because individual efficiencies are difficult to determine. These two efficiencies account for mechanical loss between the engine output shaft and thrust efficiency by the propeller [25].

2.2.2 All-Electric Variant

In this variant (Figure 2-4) an electric motor, motor controller, an electronics bus with a battery management system (BMS), and a battery pack replace the gas engine and fuel tank. The electric motor will provide power directly to the propeller and the power output is controlled by the motor controller, which changes the rotation speed to control power output. The battery pack stores and supplies the power used by the main electric motor, and the power distribution is controlled by the electronics bus. Other components that operate on electricity will not be considered since the primary power draw will be the electric motor driving the

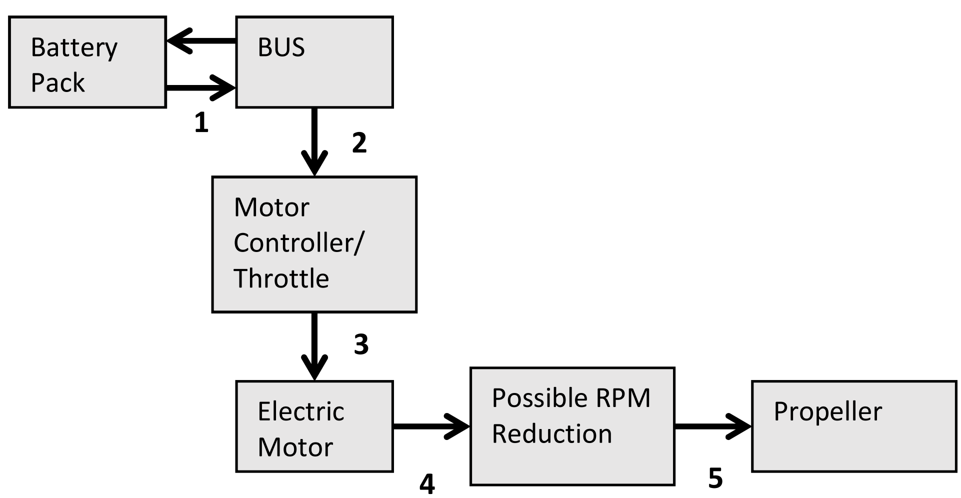

Figure 2-4: All-electric system diagram showing power flow from battery energy storage to the propeller.

propeller. Existing aircraft have an electrical bus in some manner drawing power from an alternator or on-board battery to operate avionics, lights, and control surfaces.

Table 2-2 lists the efficiencies for energy transfer between blocks designated in Figure 2-4. Efficiencies 2 and 3 are grouped together and are represented by one value since in-depth design and analysis of the bus and motor controller would be needed to determine the individual values [16]. Efficiency 4 represents the electric motor efficiency with motor manufacturers quoting greater than 0.95 [26]. Efficiency 5 is any possible loss due to RPM reduction and propeller efficiency.

Table 2-2: Efficiencies for All-Electric System

Label	Efficiency
1	Dependent on C-rate
2 & 3	0.95
4	0.96
5	0.8

Efficiency 1 is dependent on a battery cell property called the C-rate and is the ratio of discharge amps to the amp-hours of the battery (the battery capacity). Increasing the C-rate decreases the efficiency of energy provided by the cell. A faster discharge rate increases the energy lost to the battery's internal resistance as heat. Decreasing the C-rate can be accomplished by a lower power demand or by increasing the size of the battery pack so less energy is needed at any given instance from the battery pack. These same ideas are relevant for the hybrid variant as well.

2.2.3 Hybrid-Electric Variant

The hybrid-electric system is a serial hybrid system that is similar to the electric engine variant, but adds the ability generate electricity to power the electric motor and to possibly charge the batteries. The serial system utilizes batteries, but the battery pack will be sized such that the battery pack and electricity generator in tandem provide sufficient energy for conditions requiring maximum power.

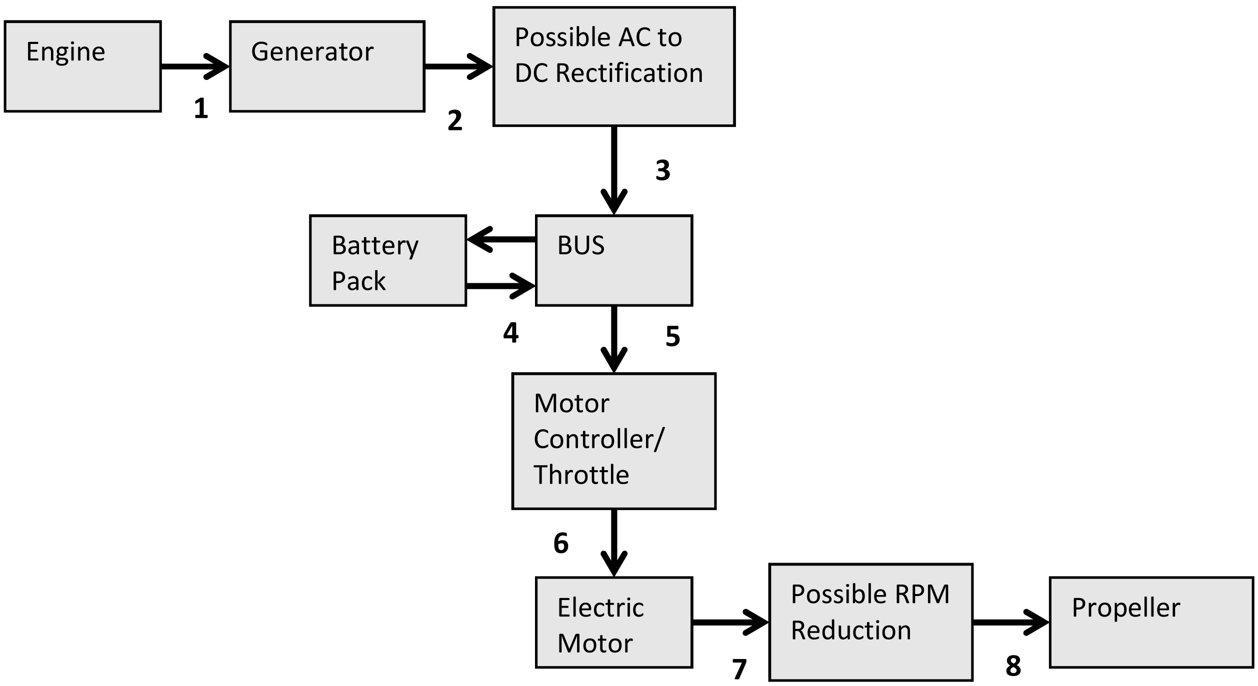

Figure 2-5: Serial hybrid system diagram showing the general components to transmit stored or generated energy to the propeller.

Table 2-3: Efficiencies for serial hybrid system	
Label	Efficiency
1 & 2	0.3
3 & 5 & 6	0.9
4	Dependent on C-rate
7	0.96
8	0.9

The efficiencies for this system are similar to the electric variant, now with the added combustion engine efficiencies. As noted previously, combustion engines will lose useful energy to thermal energy and internal mechanical losses resulting in an efficiency of 0.3 [25]. In the model this will be the efficiency of converting fuel into electric power and is represented by Arrow 1 & 2 in Figure 2-5. A key difference in the operating characteristics of the generator engine and the gas-variant power plant engine is the generator engine operates under a low and constant load, and will not need to constantly increase or decrease in RPM during the flight. Relative to the all-electric variant, the serial-hybrid system is supplementing the power stored in the batteries, with the goal to enable longer range, inflight charging, and ground charging when access to an electrical outlet is not available.

2.3 Aircraft

The aircraft used in this analysis are a Diamond DA40, a Cessna Skyhawk 172S, and a Cirrus SR22, which are all single engine aircraft that can be flown by pilots with a private pilot rating and are not considered higher powered aircraft. The DA40 and C172 are both similar weights with similar amounts of rated engine horsepower while the SR22 is a larger, more powerful aircraft.

The relevant values listed in Table 2-4, Table 2-5, and Table 2-6 were derived using different approaches. The simplest was referring to the plane's Pilot's Operating Handbook (POH) and either directly using a value or deriving the value using a simple calculation. A second method, specifically for estimating surface area, was to measure the drawings in the POH and scaling up the dimension to real life dimensions. (The POH drawings all provided basic length, wingspan, and height dimensions, which were used to determine the scaling factor.) A third method used estimated values obtained from relevant literature and an aircraft design textbook, *Aircraft Design: A Conceptual Approach* [16].

2.3.1 Diamond DA40

The first aircraft used in the model is Diamond Aircraft's DA40 [27], which is a 4-seater, single engine aircraft. The propeller is a constant speed propeller that is powered by a 134kW (180HP) Lycoming IO-360M1-A engine [28]. Table 2-4 lists various values about the aircraft that are used in the model. major characteristics to note about the aircraft is that it has a large aspect ratio, is composite construction, and is a more modern design that also resembles a glider.

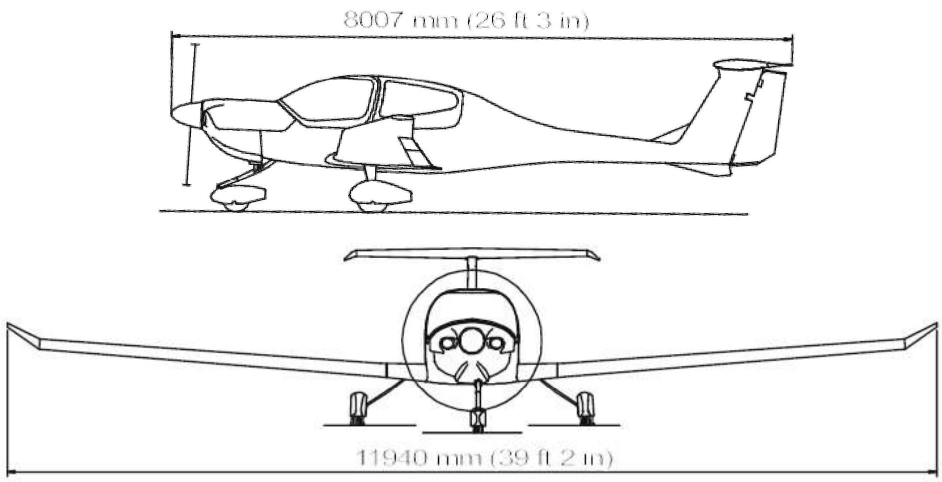

Figure 2-6: A sketch in the POH showing the overall diminsions of the Diamond DA40.

In Table 2-4, Table 2-5, and Table 2-6 are important speeds for all three aircraft that are listed as "KIAS," and is a shorthand for "Knots Indicated Airspeed." Indicated Airspeed (IAS) is the airspeed that is shown by the airspeed indicator on the instrument panel inside the aircraft. While IAS is important to aircraft operation, different speeds, based off of IAS, are used for analysis and are discussed in detail in Section 3.1. In the table V_x is the best-climb speed for shortest horizontal travel, V_y is best-climb speed for shortest time-to-climb, V_S is the stall speed, V_A is the maneuvering speed, and V_{cruise} is the cruise speed.

Table 2-4: Selected data for DA40	
Engine weight, kg (lb)	136.1 (300)
Power, kW (hp)	134.2 (180)
Fuel Type	100 LL AvGas
Total Fuel Quantity, L (gal)	156 (41.2)
Wing Area, m2 (ft2)	13.24 (113.0)
Aircraft Mass, kg (lb)	1150 (2535)
Propeller Diameter, m (in)	1.8 (74.8)
Aspect Ratio	10.5
Horizontal Tail Area, m2 (ft2)	2.3 (25.2)
Vertical Tail Area, m2 (ft2)	1.6 (17.2)
Wetted Area, m2 (ft2)	63.0 (667.8)
V_A, m/s (KIAS)	55.6 (108)
V_y, m/s (KIAS)	34.5 (67)
V_x, m/s (KIAS)	34.5 (67)
V_S, m/s (KIAS)	26.8 (52)
V_{cruise}, m/s (KIAS)	56.6 (110)

2.3.2 Cessna 172

The second aircraft is the Cessna 172S "Skyhawk" aircraft (C172) [29] and is a 4-seater single engine aircraft. The engine is a Lycoming IO-360-L2A rated at 134kW (180HP) [30] with a fixed pitch, 2-blade propeller. It is a high wing aircraft that is slower, and a much older design

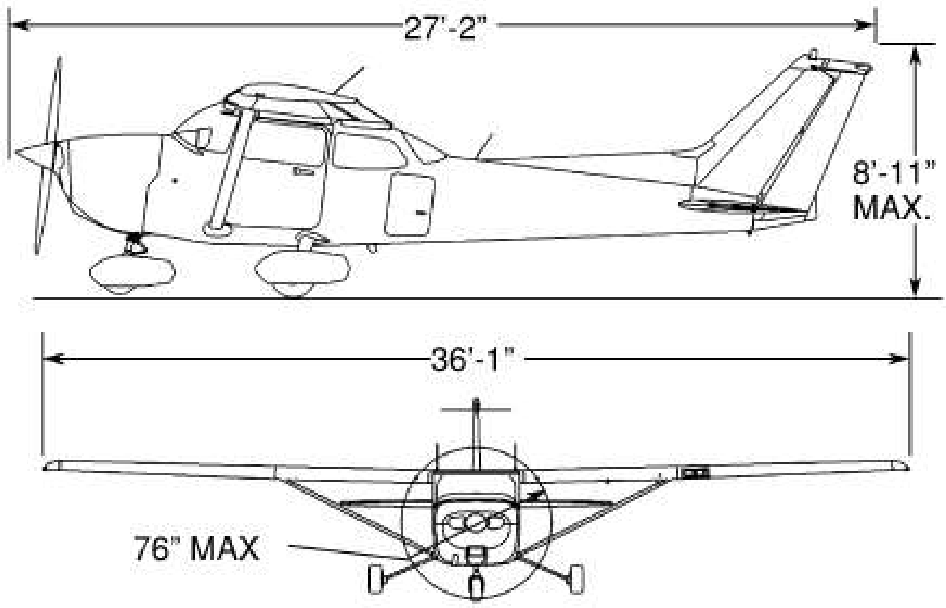

Figure 2-7: Sketch of Cessna 172S from POH showing overall dimensions and aircraft shape.

compared to the DA40 and SR22. Table 2-5 lists some relevant data of the aircraft.

The C172 is a common plane used by general aviation pilots and by flight schools for training pilots where a large majority the training flight is staying close to an airport, and often flying in a traffic pattern performing airport operations practice. This plane was analyzed in this report due to its popularity.

Table 2-5: Selected data for C172	
Engine mass, kg (lb)	136.1 (278)
Power, kW (hp)	134.2 (180)
Fuel Type	100 LL AvGas
Total Fuel Quantity, L (gal)	212 (56)
Wing Area, m^2 (ft^2)	16.2 (174)
Aircraft Mass, kg (lb)	1156 (2548)
Propeller Diameter, m (in)	1.9 (76)
Aspect Ratio	7.48
Horizontal Tail Area, m^2 (ft^2)	4.3 (46.2)
Vertical Tail Area, m^2 (ft^2)	2.6 (28.0)
Wetted Area, m^2 (ft^2)	78.4 (844.4)
V_A, m/s (KIAS)	54.0 (105)
V_y, m/s (KIAS)	38.1 (74)
V_x, m/s (KIAS)	28.8 (56)
V_S, m/s (KIAS)	27.3 (53)
V_{cruise}, m/s (KIAS)	56.6 (110)

2.3.3 Cirrus SR22

The third aircraft is Cirrus SR22 [31] which is also a 4-seater single engine aircraft. The engine is a Continental IO-550-N rated at 231kW (310HP) [32] with a constant speed, 3-blade propeller. The aircraft is a larger, more powerful airplane compared to the C172 and DA40, and some performance details are listed in Table 2-6.

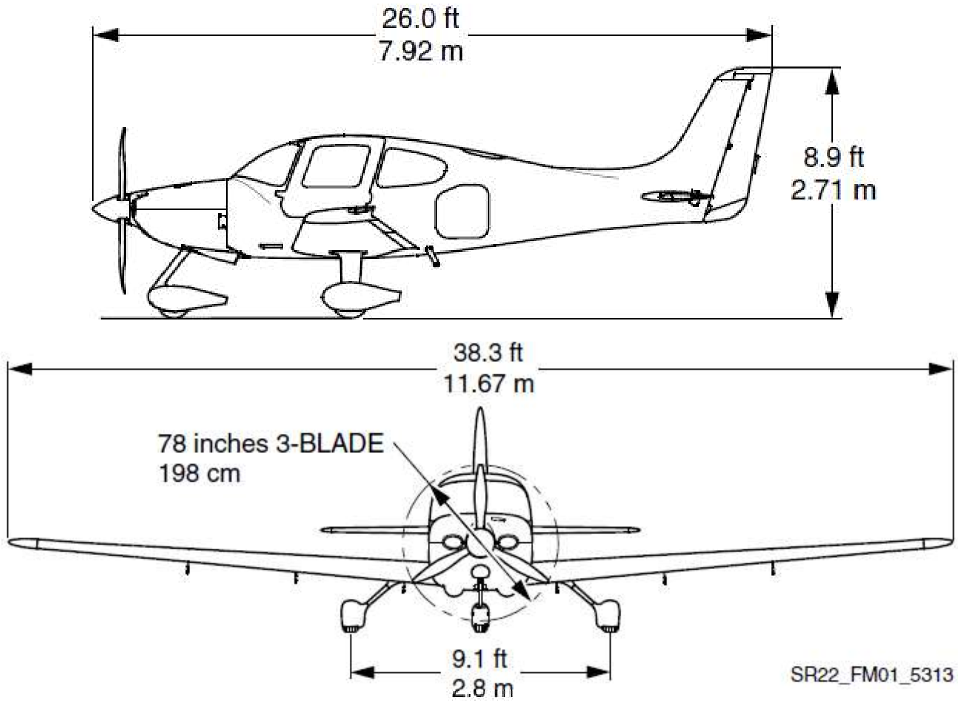

Figure 2-8: Sketch of Cirrus SR22 from POH showing overall dimensions and aircraft shape.

Table 2-6: Selected data for SR22	
Engine weight, kg (lb)	225 (496)
Power, kW (hp)	231 (310)
Fuel Type	100 LL AvGas
Total Fuel Quantity, L (gal)	358.0 (94.5)
Wing Area, m2 (ft2)	13.5 (145.2)
Aircraft Mass, kg (lb)	1633 (3600)
Propeller Diameter, cm (in)	198 (78)
Aspect Ratio	10.1
Horizontal Tail Area, m2 (ft2)	5.34 (57.5)
Vertical Tail Area, m2 (ft2)	3.32 (35.7)
Wetted Area, m2 (ft2)	61.8 (665.2)
V_A, m/s (KIAS)	55.6 (108)
V_y, m/s (KIAS)	55.6 (108)
V_x, m/s (KIAS)	45.3 (88)
V_S, m/s (KIAS)	38.1 (74)
V_{cruise}, m/s (KIAS)	77.2 (150)

2.4 Batteries

The next few sections will describe the major components used in the all-electric and hybrid-electric variants. The first important component affecting the final weight of the aircraft is the battery pack made up of individual cells. The amount of energy stored in one of these cells, the cell density, is especially important for aircraft thus a suitable cell chemistry is a lithium-polymer (LiPo) based cell. LiPo is readily available in the 18650-style cell. Basic properties of a Panasonic 18650 [33] cell are listed in Table 2-7, but other manufacturers of 18650-type battery cells will cite performance in a similar range. Battery technology is an evolving area and a potential differentiator between competitors in both the aviation industry and ground transportation industry.

Table 2-7: Stats for Panasonic NCR18650BD LiPo Battery Cell	
Voltage, V	3.6
Capacity, mAh	2980
Weight, g	49.5
Length, mm	65.10
Diameter, mm	18.25
Power Density, Wh/kg	217

2.5 Electric Motors

In the all-electric and serial-hybrid variants, an electric motor replaces the existing gas engine to directly drive the propeller. Electric motors can be classified in two broad categories: alternating current (AC) and direct current (DC) motors. AC motors can be further classified as synchronous and induction motors, while DC can be classified as brushed or brushless motors. AC or DC motors can use permeant magnets or electromagnets to operate. A major decision for the system is to choose between an AC or DC motor. An AC motor will require an inverter to convert the battery's DC power to the motor's AC power input while a DC motor can draw directly from the batteries without conversion. One reason this report focuses on DC motors is the battery packs produce direct current, and conversion to AC is not needed. A second reason is that many of the AC motors found in product catalogs have insufficient power and high weight. The DC motors found provide performance metrics for existing products, and are listed in Table 2-8 and Table 2-9. These DC motors are permanent magnet, brushless motors with lower weights, in the desired power range, and desired RPM range.

The motor properties listed are for three electric motor families made by three different manufacturers. Siemens motors are currently being used in Bye Aerospace and Pipistrel electric aircraft. (However, Siemens might be out the e-aircraft business now because they sold their property to Rolls-Royce at the end of 2019. The data listed in Table 2-8 is from 2018 and is still useful for obtaining a benchmark of existing electric motor properties [29].) YASA [30] and MagniX [26] advertise electric motors being used in development aircraft or being advertised for aerospace applications. Both companies are also advertising motor controllers and this report assumes the weight is included in the power plant weight, W_{PP} (Eq. 3-32), discussed later.

Motors specified in product catalogs have an operating voltage and a maximum load, and when multiplied together, result in the motor's power. The combination of voltage and amps correspond to a torque and a specific RPM. For an aircraft, propellers have a structural limit and lose efficiency at high rotational speeds. To stay below the propeller's speed limit, a gear reduction system is used to allow for a mismatch between engine/motor and the propeller.

The voltage and current of the motor are factors that influence the number of battery cells. DC power sources, such as battery cells, add voltage when in series and add current when in parallel. Battery cells are rated in amp-hours (Ah) and are a measure of the amount of stored energy. Amp-hours increase with the number of cells, which increases the aircraft's endurance. A single cell will produce a few amps and a few volts and linking battery cells together achieves the required voltage and amperage to operate the electric motor.

Table 2-8: Siemens Electric Motors

	SP70D	SP55D	SP260D	SP200D
Motor Volts, V	400	400	580	580
Motor Max Power, kW (HP)	92 (123)	72 (97)	260 (347)	204 (274)
Motor Cont. Power, kW (HP)	70 (94)	55 (74)	260 (347)	204 (274)
Motor Max Torque, Nm	340	240	977	1500
Motor Cont. Torque, Nm	260	180	1000	1500
Motor, RPM	2600	3000	2500	1300
Peak efficiency	0.95	0.95	0.95	0.95
Weight, kg (lb)	26 (57)	26 (57)	50 (110)	49 (108)

Table 2-9: Yasa and MagniX DC Electric Motors

	YASA 400	YASA 750	magni250
Motor Volts, V	700	350 or 700	--
Motor Max Power, kW (HP)	160 (215)	100 or 200 (134 or 268)	--
Motor Cont. Power, kW (HP)	100 (134)	70 (94)	280 (375)
Motor Max Torque, Nm	--	--	--
Motor Cont. Torque, Nm	--	--	1407
Motor, RPM	8000	3250	1900
Peak efficiency	0.96	0.96	>0.93
Weight, kg (lb)	24 (53)	37 (82)	71 (157)

2.6 Gas Engines

The hybrid-electric variant uses a gas engine to generate electricity from onboard fuel. The power requirements for this engine are lower since the engine will receive supplemental power from the batteries for instances where the main electric motor needs full power. The power range required is available in existing light-sport aircraft engines, and these engines are already designed to be light-weight and fit in compact spaces. A list of existing engines are given in Table 2-10 and the weight and power data will be used in the model described in Section 3. [34–38]

Table 2-10: Light-Sport Aircraft Engines

Engine	Weight, kg (lb)	Power, kW (hp)
Jabiru 3300	81 (178)	89 (120)
Rotax 503 UL	47 (103)	37 (50)
Rotax 582 UL	49 (108)	48 (65)
Rotax 912 A/F/UL	61 (134)	60 (81)
Rotax 912 S/ULS	64 (141)	75 (100)
Rotax 914 F/UL	76 (167)	86 (115)
Power plant Dynamics Gemini 100	87 (191)	75 (100)
Teledyne Continental O-200D	77 (170)	75 (100)
Wilksch Airmotive WAM-100	119 (262)	75 (100)
Wilksch Airmotive WAM-120	127 (280)	89 (120)

2.7 Electricity Generators

The hybrid system still needs a way to convert the mechanical energy from the generator engine into electricity, and an electricity generator will serve this function in the system diagram. An electric generator already exists on airplanes (and cars) today as an alternator, but an alternator is designed for low-power usage such as charging lead-acid batteries and powering onboard electronics. This means the power output is lower than needed for a hybrid-electric system. A serial-hybrid system requires an electric generator that can provide larger amounts of power.

For the purposes of estimating weight, an electric motor will be used as a starting point. An electric motor used “backwards” functions like a generator, so a second electric motor in the system can convert the rotational mechanical energy from the generator engine into DC current. This is how regenerative braking works in battery electric and hybrid vehicles; the motor stops using energy and instead converts some rotational energy of the wheels back into electrical energy and returned to the system. An interesting claim by Pipistrel is that pattern practice can regenerate up to 17% system energy by the main electric motor functioning as a windmill on descent [39].

2.8 Other Power Draws

The electric motor turning the propeller will be the primary component consuming power, but some power is needed for communications and avionics. Current avionics typically operate at around 24V instead of the minimum of 400V the electric motor will require. This report will not account for avionics power usage as it will be small compared to the electric motor and avionics packages can be different between aircraft families and individual aircraft of the exact same type.

3 Calculation Framework

This section outlines the model developed and used to estimate the performance of an aircraft, and relies significantly on the textbook, *Aircraft Design: A Conceptual Approach* by Daniel P. Raymer for relationships, general trends and some assumptions [16]. The performance calculations will be performed essentially two times with the first instance analyzing the aircraft using an average power-to-weight ratio of the electric motors and generator engines, to gain insight into general trends using currently available electric aircraft technology. A second iteration will be done by selecting a specific electric motor and generator engine to account for restrictions in current technology such as specific sizes of engines and motors.

3.1 Note About Airspeeds, Atmosphere

An IACO atmosphere is used for standard pressure and density values to calculate thrust and velocities. At a given altitude other than sea level, the aircraft's actual velocity changes relative to what is indicated inside the aircraft. There are four types of airspeeds often stated and are relevant to designers and pilots: Indicated airspeed (IAS), calibrated airspeed (CAS), equivalent airspeed (EAS), and true airspeed (TAS). Indicated airspeed is the speed that is displayed by the airspeed indicator on the instrument panel of the aircraft. Calibrated airspeed accounts any inaccuracies in the IAS gage itself or in the pitot static system. Equivalent airspeed accounts for airspeed differences due to speed and the effect of air compressibility in the pitot tube. True airspeed is a function of EAS and accounts for air density differences between sea-level and a given altitude. The trend is that TAS is higher than IAS at higher altitudes and the difference increases as altitude increases.

The velocities used in the following calculation are the TAS, but quoted airspeeds will be IAS since this is the speed indicated in the aircraft. The two will be clearly differentiated. IAS and CAS will be assumed to be the same, but calibration factors are available in an aircraft's POH. EAS can be determined from CAS using Eq. 3-1 and then TAS can be estimated with Eq. 3-2. The equation terms are defined as follows: p and p_o are pressure at a given altitude and sea-level pressure respectively, ρ and ρ_o are the air density at a given altitude and sea-level air density respectively, M is the Mach number, SoS is the speed of sound at a given altitude.

$$EAS = CAS\sqrt{\frac{p}{p_o}}\left(\frac{\left(\frac{q_c}{p}+1\right)^{\frac{2}{7}}-1}{\left(\frac{q_c}{p_o}+1\right)^{\frac{2}{7}}-1}\right)^{0.5} \qquad \text{Eq. 3-1}$$

With $q_c = p((1+0.2M^2)^{3.5}-1)$

And $M = \frac{TAS}{SoS}$

$$TAS = \frac{EAS}{\sqrt{\frac{\rho}{\rho_o}}} \qquad \text{Eq. 3-2}$$

3.2 Estimating Aircraft Thrust, Drag, and Lift

This section will describe the equations and any assumptions needed to determine the required thrust during the climb, cruise, and descent phases of flight. Climb will be the most power intensive stage, using 100% thrust and cruise will use a portion of the full thrust to maintain level flight. Descent will have the benefit of converting potential energy (altitude) into forward movement (velocity), meaning the cruise speed can be maintained with even less thrust.

Thrust, drag, lift and weight are four forces acting on an aircraft while in flight and need to be determined before calculating power requirements. A free-body diagram in Figure 3-1 shows an aircraft in an arbitrary orientation with axes designated as follows: x denotes the longitudinal axis of the aircraft and the y axis is perpendicular to the x axis. x’ denotes the horizon and y’ is perpendicular to x’. When the aircraft pitches up or down, the angle between x and x’ is the pitch angle (γ) and referred to as the climb angle or descent angle depending on the phase. Thrust (T) is generated by the propulsion system, in the same direction as the velocity, and opposed by drag and in the opposite direction. Lift (L) is always perpendicular to the longitudinal axis of the aircraft since it is generated by the wings and is mostly opposed by Weight (W) which is always pointing towards the Earth, opposite to the y' direction.

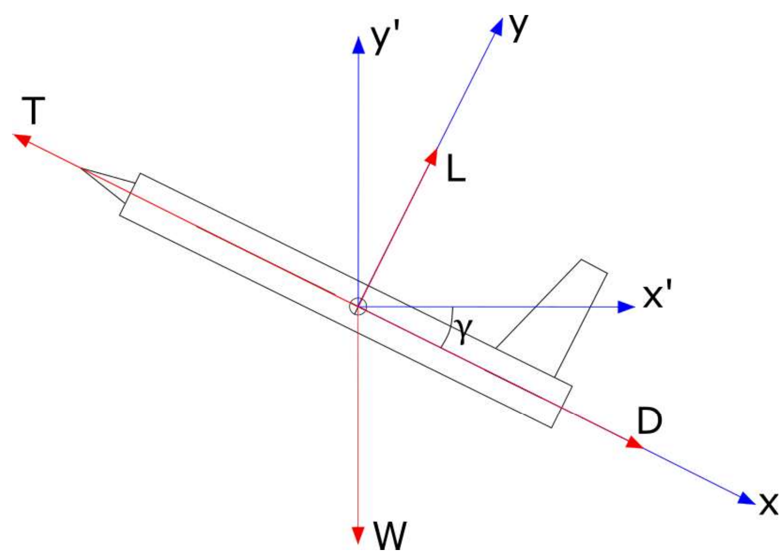

Figure 3-1: Free body diagram of an aircraft at some arbitrary flight orientation.

Summation of all the forces leads to Eq. 3-3 and Eq. 3-4.

$$\sum F_x = T - D - Wsin(\gamma) \qquad \text{Eq. 3-3}$$

$$\sum F_y = L - Wcos(\gamma) \qquad \text{Eq. 3-4}$$

During a steady climb, the aircraft is traveling at a constant speed and a constant climb angle which means the sum of the forces equals zero and Eq. 3-3 and Eq. 3-4 can be rearranged to determine thrust and lift in Eq. 3-5 and Eq. 3-6.

$$T = D + Wsin(\gamma) \qquad \text{Eq. 3-5}$$

$$L = Wcos(\gamma) \qquad \text{Eq. 3-6}$$

Weight is the max gross weight of the aircraft, the worst-case scenario for thrust requirements, and any aircraft at a lower gross weight can climb faster with equivalent amounts of thrust. Weight is simply the mass of the aircraft multiplied by the acceleration of gravity. Drag is calculated using Eq. 3-7 and is a function of air density (ρ) , velocity (V), surface area (S) and the drag coefficient (C_{D_o}). The drag coefficient requires more explanation and is described in the following paragraphs.

$$D = \frac{1}{2} C_{D_o} \rho S V^2 \qquad \text{Eq. 3-7}$$

The drag force, specifically the drag coefficient, has two main components: parasite drag and induced drag. Parasite drag is the drag associated with skin friction and other components which do not strongly correlate with lift (e.g. drag around the fuselage, landing gear, struts). Induced drag is a drag caused by the wings generating lift and it is a function of the coefficient of lift. Eq. 3-8 approximates the induced drag, which relates the suction force (S_f) along the leading edge and surface of the airfoil to the wing's angle of attack (α). The airflow over the wing shape will "detach" and become turbulent at some point along the airfoil curve, and is represented by the K factor in Eq. 3-9. When K_o is larger than K_{100}, the airfoil is generating turbulent airflow and causing more drag. In steady flight, values between 0.85 and 0.95 will be used. [16]

$$C_{D_I} = KC_L^2 \qquad \text{Eq. 3-8}$$

$$K = S_f K_{100} - (1 - S_f) K_0 \qquad \text{Eq. 3-9}$$

$$K_0 = \frac{1}{C_{L_\alpha}}$$

$$K_{100} = \frac{1}{\pi A}$$

The term, C_L, is the coefficient of lift for a 3D airfoil and C_{L_α} is the slope of the C_L vs angle-of-attack curve for a given wing. The coefficient of lift for a 3D airfoil at any angle of attack is a function of the 2D airfoil's coefficient of lift slope, C_{ℓ_α}, and calculated using Eq. 3-10. C_{ℓ_α} is available in charts that show how the coefficient of lift changes depending on the angle of attack. Referencing airfoil data curves for each aircraft, C_{ℓ_α} is chosen at 10 degrees angle of attack [40–42]. To simplify the analysis and due to the difficultly in determining the angle of attack at level flight, angle of attack is going to be the same as climb angle. For the purposes of this report, this will be a sufficient estimate.

$$C_{L_\alpha} = \left(\frac{2\pi A}{2 + \sqrt{4 + \frac{A^2\beta^2}{\eta^2}\left(1 + \frac{\tan^2 \Lambda}{\beta^2}\right)}} \right) 0.98 \qquad \text{Eq. 3-10}$$

$$\beta^2 = 1 - M^2 \qquad \text{Eq. 3-11}$$

$$\eta = \frac{\beta C_{\ell_\alpha}}{2\pi} \qquad \text{Eq. 3-12}$$

The other drag component, parasite drag, C_{D_p}, can be estimated with Eq. 3-13 where C_{fe} is a constant that changes depending on the class of aircraft. For a general aviation aircraft $C_{fe} = 0.0055$ [16]. Eq. 3-13 provides a sufficient estimate for this report without going into detail analyzing the drag created by various components of the aircraft.

$$C_{D_p} = C_{fe}\frac{Swet}{Sref} \qquad \text{Eq. 3-13}$$

The terms S_{wet} and S_{ref} are the wetted surface area and the wing reference surface area, which are listed in Table 2-4, Table 2-5, and Table 2-6. The wing and tail areas in the tables are areas in a 2D plane and not the surface areas, thus the surface area is obtained by doubling the area to account for a top and bottom surface. Adding all surface areas, including wing surface area, tail surface area, and fuselage area, together result in S_{wet} while the surface area of only the wing is S_{ref}.

With both drag coefficient components, the overall drag coefficient is $C_{D_o} = C_{D_I} + C_{D_p}$ and used to calculate drag with Eq. 3-7. Now that drag is known, thrust can be calculated as well. The next step determines the aircraft's required power output and the process is explained in Sections 3.2.1 and 3.2.2.

3.2.1 Propeller Analysis

Two methods can be used to relate thrust to engine power. Raymer presents and claims Eq. 3-14 is sufficient to determine power requirements from the required thrust while the alternate method, actuator disk theory, is more appropriate for propeller designers instead of actual engine selection for an aircraft. Actuator disk theory is outlined in 3.2.2 for completeness, but power requirements and performance characteristics will be based on the power relationship in this section [16].

With thrust determined, power required by the engine can be calculated using Eq. 3-14 where the propeller efficiency, η_p, accounts for the losses of energy due to propeller design and any losses between the engine output shaft and propeller aerodynamic losses.

$$P = \frac{TV}{\eta_p} \qquad \text{Eq. 3-14}$$

3.2.2 Actuator Disk Method

Actuator disk theory is where a “magic disk” representing a propeller increases air spread after air passes through the disk and the force required to accelerate the air is the thrust. Incoming air velocity continuously increases before the disk and increases more behind the actuator disk while the pressure decreases before the propeller, reaches a discontinuity, then decreases behind the propeller, as illustrated in Figure 3-3.

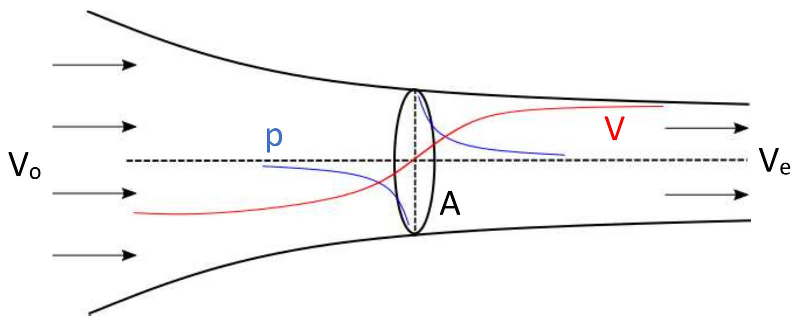

Figure 3-3: Actuator disk slip stream, pressure and velocity profile.

Half of the total velocity change, ΔV , occurs on each side of the actuator disk, and the pressure immediately before and after the disk lead to equation Eq. 3-15.

$$\mathrm{F} = \Delta \mathrm{pA} \quad \text{Eq. 3-15}$$

Newton’s equation (Eq. 3-16) for fluid flow must equal rate of change of momentum for the control volume that the disk is acting on. The terms ρQ together represent the mass flowrate of the air through the actuator disk where ρ is air density and Q is the flow rate.

$$\mathrm{F} = \dot{\mathrm{m}}\mathrm{a} = \rho \mathrm{Q} \Delta V \quad \text{Eq. 3-16}$$

Using Bernoulli’s equation (Eq. 3-17) and defining the pressure up stream (p_u) and down stream (p_d),

$$p + \frac{1}{2}\rho V^2 + \rho g h = constant \quad \text{Eq. 3-17}$$

and picking a point upstream with velocity V_o and a point downstream the disk where $V_e = V_o + \Delta V$, allows Bernoulli’s equation to be written as $p_u + \frac{1}{2}\rho V_o^2 = p_d + \frac{1}{2}\rho(V_o + \Delta V)^2$ and rewriting the equation for Δp produces Eq. 3-18. (h is the elevation change of the fluid, but since the slipstream is straight, $h = 0$.)

$$\Delta p = \rho \Delta V \left(V_o + \frac{\Delta V}{2} \right) \qquad \text{Eq. 3-18}$$

Setting Eq. 3-15 and Eq. 3-16 equal and solving for Δp allows for the expression $\frac{\rho}{A} Q \Delta V = \rho \Delta V \left(V_o + \frac{\Delta V}{2} \right)$ when combined with Eq. 3-18 . Solving for Q, results in Eq. 3-19.

$$Q = A \left(V_o + \frac{\Delta V}{2} \right) = A \left(\frac{V_o + V_e}{2} \right) \qquad \text{Eq. 3-19}$$

Eq. 3-16 can be updated by inserting Eq. 3-19.

$$F = \rho A \left(V_o + \frac{\Delta V}{2} \right) \Delta V = \frac{\rho A}{2} (V_e^2 - V_o^2) \qquad \text{Eq. 3-20}$$

Eq. 3-20 is the amount of thrust needed to speed up the air in the slipstream and is the thrust that the propeller needs to produce. To estimate the power, P_{out}, that is generated by the thrust

$$P_{out} = F V_o = \frac{\rho A V_o}{2} (V_e^2 - V_o^2) \qquad \text{Eq. 3-21}$$

The power exerted by the actuator disk, P_{in}, is half of the total ΔV between the airspeed and exhaust velocity and multiplied by the thrust.

$$P_{in} = F \frac{V_o + V_e}{2} \qquad \text{Eq. 3-22}$$

The efficiency of the actuator disk is then simply

$$\epsilon_{AD} = \frac{P_{out}}{P_{in}} \qquad \text{Eq. 3-23}$$

3.2.3 Climb

A sustained climb requires the most power since an aircraft at a designated climb angle γ is acting against both gravity and drag. Weight is the x-direction component, and drag, operating opposite thrust, is a function of the climb speed. The vertical component of the climb speed, the vertical velocity, is another useful parameter for performance comparisons because it indicates how long the aircraft will spend ascending to a desired altitude.

3.2.4 Cruise

At cruise, the free-body equations simplify to $L = W$ and $T = D$. While the aircraft's speed is faster than in the climb phase, power will be a fraction of the full throttle due to less air resistance and the horizontal weight component equal to zero. The desired cruise altitude used for analysis is 2438m (8000ft).

3.2.5 Descent

The descent phase of the flight is where the aircraft is still flying at cruising speed but is descending from the cruise altitude to the desired destination. Descent is modeled as a controlled vertical velocity, represented by a negative value, and is a simple sine relationship to indicated airspeed as in Eq. 3-24.

$$V_v = V \sin \gamma_{descent} \qquad \text{Eq. 3-24}$$

The thrust needed to maintain a desired indicated airspeed can be determined again using Eq. 3-5 but with a negative γ, subtracting the horizontal weight component from the drag force, reducing the needed thrust. Since weight is a large factor in many of these equations, estimating weight of the aircraft and of the power plant is discussed next.

3.3 Weight Estimation

A typical weight profile for the aircraft (at its current rated maximum weight) is listed Table 3-1. In every POH, there is a weight and balance section, specifically intended for a pilot to correctly load the aircraft. One of the line items is the Basic Empty Weight (BEW) and is essentially "fixed," but the pilot has control over fuel, passengers, and cargo.

Table 3-1: Example of a Typical Weight Breakdown	
Component	Mass, kg (lb)
BEW	900 (1984)
Usable Fuel	76 (168)
Pilot and Front Seat Passenger	150 (331)
Rear Seat Passengers	145 (320)
Cargo	7 (17)
Total	1310 (2888)

Fuel weight can be estimated as 6lbs for every gallon of AvGas, and is consumed during flight, converted into mechanical energy, heat, and gaseous combustion products by the gas engine. The net effect is the weight reduces as the motor is operating. In contrast, an all-electric aircraft with batteries does not change weight throughout flight because energy is produced with the movement of electrons inside the batteries, but this mass does not leave the aircraft. A hybrid aircraft is a combination of the two systems. Some mass loss will occur due to fuel consumption, but the effect will be less pronounced due to the fixed battery mass and less onboard fuel.

3.3.1 Basic Empty Weight

Basic Empty Weight is the standard aircraft weight that includes hydraulic fluid, unusable fuel, cooling and lubricating oil, and optional equipment that is not intended to be removed between flights [43]. The max gross weight is the maximum weight an aircraft can have at takeoff and is the limit to the sum of the BEW, pilot and passengers, fuel, and cargo.

The following equations and variables are taken directly from *Aircraft Design: A Conceptual Approach* [16] and are statically-derived equations based on historical and existing aircraft. These equations help to describe and estimate individual components of existing aircraft and are intended for existing single engine aircraft operating on fuel.

$$W_{wing} = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{\cos^2 \Lambda}\right)^{0.6} q^{0.006} \lambda^{0.04} \times \left(\frac{100\, t/c}{\cos^{-0.3} \Lambda}\right) \left(N_z W_{dg}\right)^{0.49} \quad \text{Eq. 3-25}$$

$$W_{ht} = 0.016\left(N_z W_{dg}\right)^{0.414} q^{0.168} S_{ht}^{0.896} \times \left(\frac{100\, t/c}{cos^{-0.12} \Lambda_{ht}}\right)\left(\frac{A}{cos^2 \Lambda_{ht}}\right)^{0.043} \lambda_h^{-0.02} \quad \text{Eq. 3-26}$$

$$W_{vt} = 0.073\left(1 + 0.2(H_t H_v)\right)\left(N_z W_{dg}\right)^{0.376} q^{0.122} S_{vt}^{0.873} \times \left(\frac{100\, t/c}{cos^{-0.49} \Lambda_{vt}}\right)\left(\frac{A}{cos^2 \Lambda_{vt}}\right)^{0.357} \lambda_{vt}^{0.039} \quad \text{Eq. 3-27}$$

$$W_{fuselage} = 0.052\, S_f^{1.086}\left(N_z W_{dg}\right)^{0.177} L_t^{-0.051} \times \left(\frac{L}{D}\right)^{-0.072} q^{0.241} + W_{press} \quad \text{Eq. 3-28}$$

$$W_{mainLG} = 0.095(N_I W_I)^{0.768}\left(\frac{L_m}{12}\right)^{0.409} \quad \text{Eq. 3-29}$$

$$W_{noseLG} = 0.125(N_I W_I)^{0.566}\left(\frac{L_m}{12}\right)^{0.845} \quad \text{Eq. 3-30}$$

$$W_{LG} = (W_{mainLG} + W_{noseLG}) - 0.014(W_{mainLG} + W_{noseLG}) \quad \text{Eq. 3-31}$$

$$W_{PP} = 2.575\, W_{en}^{0.922} N_{en} \quad \text{Eq. 3-32}$$

$$W_{fs} = 2.49 V_t^{0.726}\left(\frac{1}{1 + \frac{V_i}{V_t}}\right)^{0.363} N_t^{0.242} N_{en}^{0.157} \quad \text{Eq. 3-33}$$

$$W_{fc} = 0.053\, L^{1.536}\, B_w^{0.371}\left(N_z W_{dg} \times 10^{-4}\right)^{0.80} \quad \text{Eq. 3-34}$$

$$W_{hyd} = K_h W_{dg}^{0.8} M^{0.5} \quad \text{Eq. 3-35}$$

$$W_{avionics} = 2.117 W_{uav}^{0.933} \quad \text{Eq. 3-36}$$

$$W_{electrical} = 12.57\left(W_{fs} + W_{avionics}\right)^{0.51} \quad \text{Eq. 3-37}$$

$$W_{air\ condition\ and\ anti-ice} = 0.265\ W_{dg}^{0.52}\ N_p^{0.68}\ W_{avionics}^{0.17}\ M^{0.08} \qquad \text{Eq. 3-38}$$

$$W_{furnishings} = 0.0582 W_{dg} - 65 \qquad \text{Eq. 3-39}$$

$$W_{wing} = 0.036 S_w^{0.758}\ W_{fw}^{0.0035} \left(\frac{A}{\cos^2 \Lambda}\right)^{0.6} \times q^{0.006} \lambda^{0.04} \left(\frac{100\ t/c}{\cos^{-0.3} \Lambda}\right) \left(N_z W_{dg}\right)^{0.49} \qquad \text{Eq. 3-40}$$

A = aspect ratio

B_w = wingspan, ft

D = fuselage structural depth

K_h = 0.05 for low subsonic with hydraulics for brakes and retracts only

L = fuselage structural length, ft

L_m = extended length of main landing gear, in

L_t = tail length; wing quarter-MAC to tail quarter-MAC, ft

M = Mach number (design maximum)

P_Δ = cabin pressure differential, typically 8psi

S_f = fuselage area, ft^2

S_{ht} = horizontal tail area, ft^2

S_{vt} = vertical tail area, ft^2

S_w = trapezoidal wing area, ft^2

V_i = integral tanks volume, gal

V_{pr} = volume of pressurized section

W_{en} = engine weight, lb

N_t = number of fuel tanks

V_t = total fuel volume, gal

W_{fw} = weight of fuel in wing, if zero ignore, lb

W_{dg} = flight design gross weight, lb

W_l = landing design gross weight, lb

W_{press} = weight penalty due to pressurization.

W_{uav} = uninstalled avionics weight, lb (typically = 800 to 1400lb), see Table 11.6 in [16]

H_t = horizontal tail height above fuselage

H_v = vertical tail height above fuselage

$H_t H_v$ = 0 for conventional tail, 1.0 for T tail

Λ = sweep angle

Λ_{ht} = 0, horizontal tail sweep angle

Λ_{vt} = 0, vertical tail sweep angle

q = dynamic pressure at cruise

λ = taper ratio

λ_h = taper ratio for tail

λ_{vt} = taper ratio (for vert tail if less than 0.2, use 0.2)

t/c = thickness to chord ratio, use average

N_{en} = number of engines (total for aircraft)

N_l = ultimate landing load factor = N_{gear} x 1.5

N_p = number of personnel onboard (crew and passengers)

N_z = ultimate load factor, = 1.5 x limit load factor

3.3.2 Electric Motor Weight and Generator Motor Weight

In Section 3.3.1, W_{PP} estimates the weight of the engine, the propeller, and any additional components that incorporate the propulsions system into the airplane. W_{PP} is a function of W_{en} and is different for each variant. In the gas variant, W_{en} is the dry weight of the engine by itself. In the all-electric variant W_{en} is the dry weight of the electric motor by itself. In the serial-hybrid variant W_{en} is the dry weight of the electric motor, electric generator, and the generator engine added together. (Reference Table 2-4, Table 2-5, and Table 2-6 for engine-only weights.)

To estimate the serial hybrid motor system weight, first we determine the generator engine's (P_{gen}), power output by dividing the required power by the total system efficiency (η_{sys}) from the output of the power source to the output shaft of the main motor, which is found by multiplying . For example, the efficiency of the serial-hybrid system is the efficiencies of arrows 3&5&6, and 7 multiplied together.

$$P_{gen} = \frac{P_{needed}}{\eta_{sys}} \quad \text{Eq. 3-41}$$

Since power generated will not sustain full-power climb, the engine will be sized to sustain level flight at a given cruise speed. To achieve full thrust, a combination of batteries and generator engine will provide the required power for a full throttle climb.

Using the engine data listed in Table 2-10 a "power loading density" (ρ_{LS}), in units of weight per power, can be determined and when multiplied by P_{gen} provides the hybrid generator engine weight, W_{HGE}, which can be calculated using Eq. 3-42.

$$W_{HGE} = P_{gen\ engine}\ \rho_{LS} \quad \text{Eq. 3-42}$$

Referencing Figure 2-5, there is one gas engine and two electric motors with one electric motor rotating the propeller while the second functions as a generator that converts the rotational output from the generator engine into electricity. W_{gen} is sized for max power of the generator engine and W_{EM} is sized for propeller power requirements.

Adding everything together, W_{HS} provides the weight of all three motors together and when used in Eq. 3-32, produces an estimated power plant weight, which is assumed to be a sufficient estimate for the hybrid and all-electric systems. A possible benefit of an electric engine is a direct connection to the propeller and less mounting hardware to support he engine, which will reduce the BEW of both the all-electric variant and the serial-hybrid. However, for the serial hybrid there will be two additional motors/engines that need to be connected and mounted in the aircraft, adding to the BEW.

$$W_{HS} = W_{HGE} + W_{gen} + W_{EM} \quad \text{Eq. 3-43}$$

In order to determine an exact size or motor, motor data in Table 2-8 and Table 2-9 were averaged together to get an approximate power-per-mass number. This number was then multiplied by the "required power" input into the propeller to approximate the engine weight.

3.3.3 Battery Mass Fraction

Eq. 3-44 is an estimate for the battery pack weight at different phases of flight and is dependent on the power density (ρ_{batt}) of the cell, motor run time (t), and power draw (P_{used}) during that phase [16]. Battery cell is data listed in Table 2-7.

$$W_{batt} = \frac{1000\ t\ P_{used}}{\rho_{batt}\ \eta_{sys}} \qquad \text{Eq. 3-44}$$

4 Results and Discussion

Aircraft performance results are presented in this section and show how the model is predicting the performance of the three types of aircraft during the main portions of a flight. Then, weight allocation of each variant's components is presented in two sets of results: a baseline version and a fixed-parameter version. The baseline results allow system components to change based on power needs and provides an initial indication of a variant's feasibility. The fixed parameter version investigates the serial-hybrid system in more detail by fixing the power plant's generator engine weight and generator engine power output. The flight profiles described in Figure 2-1 Figure 2-2 are discussed using the model results and exploring the trade-offs of each variant. Lastly, battery recharging during flight and on the ground is discussed.

4.1 Climb Performance Results

Climb performance results as the aircraft ascends from sea-level to 2438 m (8000 ft) are shown in Table 4-1, Table 4-2, and Table 4-3. Climb performance is influenced by max gross weight and the power output of the aircraft engine, but the method of power generation does not influence climb performance. During the ascent, CAS is constant while TAS increases as the altitude increases due to decreasing atmospheric density, while the vertical velocity decreases due to less thrust and lift generated by the propeller and wings at higher altitudes. Compared to the POH values for each aircraft, the reduction in climb is less than the published values, but the total time-in-climb is in line with the quoted time-in-climb numbers in an aircraft's POH.

Table 4-1: DA40 Climb Performance							
Altitude, m (ft)	Power in by motor, kW (hp)	CAS, m/s (kts)	EAS, m/s (kts)	TAS, m/s (kts)	Vertical Velocity, m/s (fpm)	Climb Angle, deg	Time to Climb, Cumulative min
0	134.2 (180)	34.5 (67)	34.46 (67.00)	34.46 (67.00)	5.1 (1004)	8.6	0
305 (1000)	134.2 (180)	34.5 (67)	34.46 (67.00)	34.98 (68.00)	5.1 (1004)	8.3	1.0
610 (2000)	134.2 (180)	34.5 (67)	34.46 (66.99)	35.53 (69.08)	5.0 (984)	8.1	2.0
914 (3000)	134.2 (180)	34.5 (67)	34.46 (66.99)	36.00 (69.99)	4.9 (965)	7.9	3.0
1219 (4000)	134.2 (180)	34.5 (67)	34.46 (66.99)	36.61 (71.17)	4.8 (945)	7.6	4.0
1524 (5000)	134.2 (180)	34.5 (67)	34.45 (66.98)	37.12 (72.17)	4.8 (945)	7.4	5.1
1829 (6000)	134.2 (180)	34.5 (67)	34.45 (66.98)	37.66 (73.21)	4.7 (925)	7.2	6.2
2134 (7000)	134.2 (180)	34.5 (67)	34.45 (66.97)	38.26 (74.37)	4.6 (906)	6.9	7.3
2438 (8000)	134.2 (180)	34.5 (67)	34.45 (66.97)	38.87 (75.56)	4.5 (787)	6.7	8.4

Table 4-2: C172 Climb Performance							
Altitude, m (ft)	Power in by motor, kW (hp)	CAS, m/s (kts)	EAS, m/s (kts)	TAS, m/s (kts)	Vertical Velocity, m/s (fpm)	Climb Angle, deg	Time to Climb, Cumulative min
0	134.2 (180)	38.1 (74)	38.04 (73.95)	38.04 (73.95)	3.8 (748.0)	5.7	0
305 (1000)	134.2 (180)	38.1 (74)	38.04 (73.95)	38.61 (75.06)	3.7 (728.3)	5.4	1.4
610 (2000)	134.2 (180)	38.1 (74)	38.04 (73.94)	39.23 (76.25)	3.6 (708.7)	5.2	2.8
914 (3000)	134.2 (180)	38.1 (74)	38.04 (73.94)	39.74 (77.25)	3.5 (689.0)	5.0	4.3
1219 (4000)	134.2 (180)	38.1 (74)	38.03 (73.93)	40.41 (78.55)	3.4 (669.3)	4.8	5.8
1524 (5000)	134.2 (180)	38.1 (74)	38.03 (73.93)	40.98 (79.65)	3.3 (649.6)	4.6	7.3
1829 (6000)	134.2 (180)	38.1 (74)	38.03 (73.92)	41.57 (80.80)	3.2 (629.9)	4.4	8.9
2134 (7000)	134.2 (180)	38.1 (74)	38.03 (73.92)	42.23 (82.09)	3.1 (610.2)	4.2	10.5
2438 (8000)	134.2 (180)	38.1 (74)	38.02 (73.91)	42.90 (83.39)	3.0 (590.6)	4.0	12.2

Table 4-3: SR22 Climb Performance							
Altitude, m (ft)	Power in by motor, kW (hp)	CAS, m/s (kts)	EAS, m/s (kts)	TAS, m/s (kts)	Vertical Velocity, m/s (fpm)	Climb Angle, deg	Time to Climb, Cumulative min
0	231.1 (309.9)	55.56 (108)	55.56 (108.00)	55.56 (108.00)	5.3 (1043.3)	5.5	0
305 (1000)	231.2 (310.0)	55.56 (108)	55.55 (107.99)	56.39 (109.61)	5.2 (1023.6)	5.3	1.0
610 (2000)	231.1 (309.9)	55.56 (108)	55.54 (107.97)	57.28 (111.34)	5.1 (1003.9)	5.1	2.0
914 (3000)	231.1 (309.9)	55.56 (108)	55.54 (107.96)	58.03 (112.80)	5.0 (984.3)	4.9	3.0
1219 (4000)	231.1 (309.9)	55.56 (108)	55.53 (107.94)	59.00 (114.68)	4.9 (964.6)	4.7	4.0
1524 (5000)	231.1 (309.9)	55.56 (108)	55.52 (107.93)	59.82 (116.28)	4.8 (944.9)	4.6	5.1
1829 (6000)	231.1 (309.9)	55.56 (108)	55.51 (107.91)	60.68 (117.95)	4.7 (925.2)	4.4	6.2
2134 (7000)	231.1 (309.9)	55.56 (108)	55.50 (107.89)	61.64 (119.81)	4.6 (905.5)	4.3	7.3
2438 (8000)	231.1 (309.9)	55.56 (108)	55.50 (107.88)	62.61 (121.71)	4.5 (885.8)	4.1	8.4

4.2 Cruise Performance Results

The cruise phase is considered next with Table 4-4 showing performance results of each gas variant. The fuel used and the time-in-cruise describe how long the aircraft can fly before the fuel is completely consumed. Fuel consumption is in units of volume per hour, where a lower value means the engine is more efficient at converting the fuel into usable power. Flight time for the all-electric and serial-hybrid configurations will be discussed later, but flight time is reduced for these variants.

Table 4-4: Cruise Performance (existing aircraft, gas variant)

	DA40	C172	SR22
Cruise Altitude, m (ft)	2438 (8000)	2438 (8000)	2438 (8000)
CAS, m/s (kts)	56.6 (110)	57.1 (111)	77.2 (150)
EAS, m/s (kts)	56.52 (109.87)	57.19 (111.16)	76.99 (149.67)
TAS, m/s (kts)	63.77 (123.95)	64.52 (125.41)	86.87 (168.86)
Thrust, N	1113.90	1452.90	1844.50
Power in by motor, kW (hp)	88.8 (119)	117.2 (157)	200.3 (269)
Time in Cruise, hr	5.64	5.31	6.08
Fuel Used Climb, L (gal)	6.4 (1.7)	11.0 (2.9)	12.5 (3.3)
Fuel Used Cruise, L (gal)	149.5 (39.5)	201.0 (53.1)	345.2 (91.2)

4.3 Descent Performance Results

Descent is the next phase with the performance results shown in Table 4-5, Table 4-6, and Table 4-7 for each aircraft, descending from2438m (8000 ft) to sea-level. As specified, the descent rate is constant, and the CAS is constant. However, the calculated power required changes with the altitude, decreasing at lower altitudes, and is due to the wings generating more lift and the propeller producing more thrust as a result of the denser atmosphere. The total descent time is 23.2 minutes and will be useful later when total flight time is discussed.

Table 4-5: DA40 Descent Performance							
Altitude, m (ft)	Power in by motor, kW (hp)	CAS, m/s (kts)	EAS, m/s (kts)	TAS, m/s (kts)	Vertical Velocity, m/s (fpm)	Descent Angle deg	Descent rate by segment min
0	53.8 (72.1)	56.59 (110)	56.59 (110.00)	56.59 (110.00)	-1.8 (-354.3)	-1.8	0
305 (1000)	54.9 (73.6)	56.59 (110)	56.58 (109.99)	57.43 (111.64)	-1.8 (-354.3)	-1.8	2.9
610 (2000)	56.2 (75.4)	56.59 (110)	56.57 (109.97)	58.34 (113.40)	-1.8 (-354.3)	-1.7	5.8
914 (3000)	57.3 (76.8)	56.59 (110)	56.57 (109.96)	59.10 (114.88)	-1.8 (-354.3)	-1.7	8.7
1219 (4000)	58.6 (78.6)	56.59 (110)	56.56 (109.94)	60.09 (116.81)	-1.8 (-354.3)	-1.7	11.6
1524 (5000)	59.8 (80.2)	56.59 (110)	56.55 (109.92)	60.93 (118.43)	-1.8 (-354.3)	-1.7	14.5
1829 (6000)	61.0 (81.8)	56.59 (110)	56.54 (109.91)	61.80 (120.13)	-1.8 (-354.3)	-1.6	17.4
2134 (7000)	62.4 (83.7)	56.59 (110)	56.53 (109.89)	62.78 (122.03)	-1.8 (-354.3)	-1.6	20.3
2438 (8000)	63.7 (85.4)	56.59 (110)	56.52 (109.87)	63.77 (123.95)	-1.8 (-354.3)	-1.6	23.2

Table 4-6: C172 Descent Performance							
Altitude, m (ft)	Power in by motor, kW (hp)	CAS, m/s (kts)	EAS, m/s (kts)	TAS, m/s (kts)	Vertical Velocity, m/s (fpm)	Descent Angle deg	Descent rate by segment min
0	78.8 (105.7)	57.1 (111)	111.29 (57.25)	111.29 (57.25)	-1.8 (-354.3)	-1.8	0
305 (1000)	80.4 (107.8)	57.1 (111)	111.28 (57.25)	112.95 (58.11)	-1.8 (-354.3)	-1.8	2.9
610 (2000)	82.0 (110.0)	57.1 (111)	111.26 (57.24)	114.73 (59.02)	-1.8 (-354.3)	-1.7	5.8
914 (3000)	83.4 (111.8)	57.1 (111)	111.25 (57.23)	116.23 (59.79)	-1.8 (-354.3)	-1.7	8.7
1219 (4000)	85.2 (114.3)	57.1 (111)	111.23 (57.22)	118.18 (60.80)	-1.8 (-354.3)	-1.7	11.6
1524 (5000)	86.8 (116.4)	57.1 (111)	111.21 (57.21)	119.82 (61.64)	-1.8 (-354.3)	-1.7	14.5
1829 (6000)	88.4 (118.5)	57.1 (111)	111.20 (57.21)	121.54 (62.53)	-1.8 (-354.3)	-1.6	17.4
2134 (7000)	90.2 (121.0)	57.1 (111)	111.18 (57.20)	123.46 (63.51)	-1.8 (-354.3)	-1.6	20.3
2438 (8000)	92.0 (123.4)	57.1 (111)	111.16 (57.19)	125.41 (64.52)	-1.8 (-354.3)	-1.6	23.2

Table 4-7: SR22 Descent Performance							
Altitude, m (ft)	Power in by motor, kW (hp)	CAS, m/s (kts)	EAS, m/s (kts)	TAS, m/s (kts)	Vertical Velocity, m/s (fpm)	Descent Angle deg	Descent rate by segment min
0	142.6 (191.2)	150.00 (77.17)	150.00 (77.17)	150.00 (77.17)	-1.8 (-354.3)	-1.3	0
305 (1000)	145.2 (194.7)	150.00 (77.17)	149.97 (77.15)	152.21 (78.30)	-1.8 (-354.3)	-1.3	2.9
610 (2000)	148.0 (198.5)	150.00 (77.17)	149.93 (77.13)	154.60 (79.53)	-1.8 (-354.3)	-1.3	5.8
914 (3000)	150.3 (201.6)	150.00 (77.17)	149.89 (77.11)	156.61 (80.57)	-1.8 (-354.3)	-1.3	8.7
1219 (4000)	153.4 (205.7)	150.00 (77.17)	149.85 (77.09)	159.21 (81.90)	-1.8 (-354.3)	-1.2	11.6
1524 (5000)	156.0 (209.2)	150.00 (77.17)	149.81 (77.07)	161.41 (83.04)	-1.8 (-354.3)	-1.2	14.5
1829 (6000)	158.7 (212.8)	150.00 (77.17)	149.77 (77.05)	163.70 (84.21)	-1.8 (-354.3)	-1.2	17.4
2134 (7000)	161.7 (216.8)	150.00 (77.17)	149.72 (77.02)	166.26 (85.53)	-1.8 (-354.3)	-1.2	20.3
2438 (8000)	164.7 (220.9)	150.00 (77.17)	149.67 (77.00)	168.86 (86.87)	-1.8 (-354.3)	-1.2	23.2

4.4 Weight Results

Three portions of an aircraft's weight are important for comparisons: Basic Empty Weight; the total aircraft weight, with fuel, but without crew or cargo; and the weight difference between total aircraft weight, with fuel and the max gross weight. These weights are included in Table 4-8, Table 4-9, and Table 4-10 along with detailed component weights that were estimated using the equations in Section 3.3.1. Many of these individual component weights remain mostly constant between each variant of an aircraft, however, the individual components that change significantly include fuel weight (due to smaller or non-existent fuel tanks), and power plant weight. Battery pack weight and fuel weight are separate line items that change between variants and are grouped into the final total weight of the aircraft, without cargo or crew. (The minimum crew for a GA aircraft is the pilot, but the "crew" in this report will include co-pilot and passengers. For reference, United States' Centers for Disease Control lists the average men's weight to be 90 kg (198 lb) and the average women's weight to be 78 kg (171 lb) [44].)

In the baseline and fixed result tables, the descent phase is ignored and only the climb and cruise phases are considered in the weight-sizing results. This is primarily due to the climb phase requiring large amount of power compared to the other phases and because the Federal Aviation Administration (FAA) specifies flight time based only on cruise. Power needed in descent, as shown in the previous tables, is typically less demanding than cruise, which means time in the cruise phase can be “substituted” to account for the descent phase. In addition, manufacturer documentation typically only lists climb performance and aircraft flight times in the cruise configuration.

4.4.1 Baseline Weight Analysis

Baseline weight results are presented in Table 4-8, Table 4-9, and Table 4-10 and assume that any size of electric generator engine is available and that the generator engine can sustain the aircraft in cruise without additional power from a battery pack. The batteries only supplement the engine when the electric motor requires maximum power output during the climb phase of the flight.

The total installed power plant weight is classified in different categories depending on the aircraft variant, but the miscellaneous weight is simply the weight not accounted for in the dry engine weight, and includes mounting hardware, add-on components, and the propeller. Also, all weights were calculated and listed in pounds because the statistical equations in section 3.3.1 are based on data measured in pounds.

Table 4-8: DA40 Variant Weights, Baseline				
		Gas	Electric	Serial-Hybrid
Component		Weight (lb)	Weight (lb)	Weight (lb)
Wing		335	329	334
Horizontal Tail		27	27	27
Vertical Tail		44	44	44
Fuselage		157	157	157
Main Landing Gear		162	162	162
Nose Landing Gear		43	43	43
Installed Power Plant	**Electric Motor**	**--**	**88**	**87**
	Main Gas Engine	**300**	**--**	**--**
	Generator Engine	**--**	**--**	**224**
	Electric Generator	**--**	**--**	**58**
	Misc. Engine Weight	**195**	**55**	**272**
	System Total (Eq. 3-32)	**495**	**143**	**641**
Avionics		79	79	79
Fuel System		38	0	10
Flight Controls		47	47	47
Hydraulics		14	14	14
Electrical		143	117	125
Furnishings		83	83	83
Total Weight		1667	1245	1766
POH Stated BEW		1620	1620	1620
Battery Pack Weight Climb		0	283	123
Battery Pack Weight Cruise		0	989	0
Battery Pack Weight Descent		0	0	0
Fuel Weight		247	0	60
Fuel Volume (gal)		41.2	0	10
Weight (without cargo, crew/passengers)		1914	2517	1927
Aircraft max gross		2535	2535	2535
Available crew/passengers, cargo		**621**	**18**	**608**

Table 4-9: C172 Variant Weights, Baseline				
		Gas	Electric	Serial-Hybrid
Component		Weight (lb)	Weight (lb)	Weight (lb)
Wing		361	354	359
Horizontal Tail		52	52	52
Vertical Tail		61	61	61
Fuselage		237	237	237
Main Landing Gear		163	163	163
Nose Landing Gear		43	43	43
Installed Power Plant	**Electric Motor**	**--**	**88**	**88**
	Main Gas Engine	**300**	**--**	**--**
	Generator Engine	**--**	**--**	**331**
	Electric Generator	**--**	**--**	**77**
	Misc. Engine Weight	**195**	**55**	**291**
	System Total (Eq. 3-32)	**495**	**143**	**787**
Avionics		81	81	81
Fuel System		47	0	10
Flight Controls		48	48	48
Hydraulics		14	14	14
Electrical		149	118	125
Furnishings		83	83	83
Total Weight		1835	1397	2063
POH Stated BEW		1653	1653	1653
Battery Pack Weight Climb		0	406	54
Battery Pack Weight Cruise		0	1305	0
Battery Pack Weight Descent		0	0	0
Fuel Weight		336	0	60
Fuel Volume (gal)		56	0	10
Weight (without cargo, crew/passengers)		2171	3109	2178
Aircraft max gross		2549	2549	2549
Available crew/passengers, cargo		**378**	**Overweight**	**371**

<table>
<tr><td colspan="5">Table 4-10: SR22 Variant Weights, Baseline</td></tr>
<tr><td colspan="2"></td><td>Gas</td><td>Electric</td><td>Serial-Hybrid</td></tr>
<tr><td colspan="2">Component</td><td>Weight (lb)</td><td>Weight (lb)</td><td>Weight (lb)</td></tr>
<tr><td colspan="2">Wing</td><td>180</td><td>176</td><td>178</td></tr>
<tr><td colspan="2">Horizontal Tail</td><td>25</td><td>25</td><td>25</td></tr>
<tr><td colspan="2">Vertical Tail</td><td>13</td><td>13</td><td>13</td></tr>
<tr><td colspan="2">Fuselage</td><td>400</td><td>400</td><td>400</td></tr>
<tr><td colspan="2">Main Landing Gear</td><td>212</td><td>212</td><td>212</td></tr>
<tr><td colspan="2">Nose Landing Gear</td><td>53</td><td>53</td><td>53</td></tr>
<tr><td rowspan="6">Installed Power Plant</td><td>Electric Motor</td><td>--</td><td>151</td><td>151</td></tr>
<tr><td>Main Gas Engine</td><td>496</td><td>--</td><td>--</td></tr>
<tr><td>Generator Engine</td><td>--</td><td>--</td><td>491</td></tr>
<tr><td>Electric Generator</td><td>--</td><td>--</td><td>131</td></tr>
<tr><td>Misc. Engine Weight</td><td>291</td><td>110</td><td>523</td></tr>
<tr><td>System Total (Eq. 3-32)</td><td>787</td><td>263</td><td>1292</td></tr>
<tr><td colspan="2">Avionics</td><td>101</td><td>101</td><td>101</td></tr>
<tr><td colspan="2">Fuel System</td><td>69</td><td>0</td><td>8</td></tr>
<tr><td colspan="2">Flight Controls</td><td>61</td><td>61</td><td>61</td></tr>
<tr><td colspan="2">Hydraulics</td><td>18</td><td>18</td><td>18</td></tr>
<tr><td colspan="2">Electrical</td><td>173</td><td>132</td><td>138</td></tr>
<tr><td colspan="2">Furnishings</td><td>145</td><td>145</td><td>145</td></tr>
<tr><td colspan="2">Total Weight</td><td>2236</td><td>1479</td><td>2644</td></tr>
<tr><td colspan="2">POH Stated BEW</td><td>2100</td><td>2100</td><td>2100</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td colspan="2">Battery Pack Weight Climb</td><td>0</td><td>483</td><td>68</td></tr>
<tr><td colspan="2">Battery Pack Weight Cruise</td><td>0</td><td>2231</td><td>0</td></tr>
<tr><td colspan="2">Battery Pack Weight Descent</td><td>0</td><td>732</td><td>0</td></tr>
<tr><td colspan="2">Fuel Weight</td><td>567</td><td>0</td><td>60</td></tr>
<tr><td colspan="2">Fuel Volume (gal)</td><td>94.5</td><td>0</td><td>10</td></tr>
<tr><td colspan="2">Weight (without cargo, crew/passengers)</td><td>2803</td><td>4193</td><td>2772</td></tr>
<tr><td colspan="2">Aircraft max gross</td><td>3600</td><td>3600</td><td>3600</td></tr>
<tr><td colspan="2">Available crew/passengers, cargo</td><td>797</td><td>Overweight</td><td>828</td></tr>
</table>

The baseline results provide insight into the feasibility of all-electric and serial-hybrid variants of existing aircraft. First, these results show that the all-electric variation will struggle to replicate the performance of the existing aircraft for all three aircraft models primarily due to the high battery weight. An overweight aircraft affects take-off and climb performance, the balance and stability, and an overweight aircraft means the wings are insufficient for generating the lift needed for safe flight.

To reduce the battery weight the all-electric variant will need to travel at a slower speed to achieve one hour of flight time. Using the SR22 as an example, flying at 150 KIAS requires a large amount of energy or a large battery pack weight. Reducing the cruise speed to 130 KIAS instead means the battery back needs to weigh 701 kg (1546 lb), putting the weight of the aircraft, without crew or cargo, at 1591 kg (3508 lb). There is still not enough space for one average-weight adult to occupy the aircraft, so the speed will need to be reduced more, but the reduction illustrates the strong effect speed has on energy requirements, and therefore battery pack weight requirements. The trade-off for an all-electric version of an existing aircraft is that speed and range will be reduced in order to operate an all-electric propulsion system. This trade-off is present in the Panthera aircraft by Pipistrel (Table 1-1), whose projected performance, such as the cruise speed, decreases when compared to the hybrid and gas variants.

Furthermore, the serial-hybrid BEW is heavier compared to the gas variant BEW due to the added components to the propulsion system and a higher BEW reduces weight available for use by energy storage, crew, or cargo. The serial-hybrid presented in Table 4-8 contains enough energy to sustain approximately 1 hour of flight time due to the assumption that the aircraft is only carrying 37.9 (10 gal) of fuel. For the baseline results, adding fuel capacity will marginally increase the BEW while increasing the total flight time of the aircraft, but this only works up to the max gross weight limit and the amount of crew and cargo that needs to be on the aircraft. Determining the flight time that the serial-hybrid can achieve will require knowing more details about the generator engine that is used and the rate of fuel consumption. These details are known when a specific engine is selected for the generator engine.

4.4.2 Fixed Engine Analysis

Based on the baseline results, the DA40 is the most promising aircraft model for a serial-hybrid conversion, and the same calculation will be repeated, but now selecting specific electric motor and gas engine models. The fixed weight for all the components also means there is a maximum amount of power available from the electric motors and generator engine. This is different from the baseline version where the weight and available generator power scaled with performance requirements.

The engine selected is the Rotax 912 S/ULS [45] because it has detailed technical data sheets available and is representative of the selection of engines in Table 2-10. Its stats are a maximum power output of 74.6 kW (100 hp), weighs 64 kg (141 lb), and consumes 26.5 L (7 gal) of fuel at maximum power.

Table 4-11 presents these results and compares them to the serial-hybrid system of Table 4-8. Looking at the power required to sustain a cruise speed of 110 KIAS an additional battery pack is needed to supplement the power needed by the main electric motor since the generator engine can only provide up to 74.6 kW. The additional battery is listed as Battery Pack Weight Cruise (Table 4-11), and sized to operate as long as the generator engine is operating and supplementing the power produced as needed.

4.5 Flight Time Analysis

The aircraft's available power is sized such that the aircraft can fly until absolutely zero fuel and, battery energy is available. The total time (at cruise) is 1.3 hours. While this provides an absolute endurance of the aircraft, the question remains: What would a flight look like for this variant? Power consumption and weight fraction is a given during the climb phase because the battery pack is sized specifically to ensure enough power is available to climb to the cruise altitude. In addition to the battery pack, the generator engine consumes 4.2 L (1.1gal) of fuel during the climb phase. The descent phase requires 23 min to descend at 1.8 mps (354.3 fpm) down to sea-level using a power setting less than the power required during the cruise phase at the same CAS. The maximum power used is 63.7 kW (85.4 hp) and correlates to approximately 4600 rpm on the generator engine, consuming 16.1 L/hr (4.25 gal/hr), or 6.8 L (1.8 gal) of fuel total. With the climb and descent phases accounted for, the remaining fuel available for the cruise phase is 26.9 L (7.1 gal) and the Rotax engines consumes fuel at 26.5 L/hr (7 gal/hr) allowing for approximately 1 hour of flight time. Table 4-12 summarizes fuel consumption and flight time.

Table 4-11: DA40 Variant Weights, Fixed			
		Baseline Serial-Hybrid	Fixed Serial-Hybrid
Component		Weight (lb)	Weight (lb)
Wing		334	333
Horizontal Tail		27	26
Vertical Tail		44	43
Fuselage		157	150
Main Landing Gear		162	162
Nose Landing Gear		43	43
Installed Power Plant	**Electric Motor**	**87**	**108**
	Generator Engine	**224**	**141**
	Electric Generator	**58**	**53**
	Misc. Engine Weight	**272**	**196**
	System Total (Eq. 3-32)	**641**	**498**
Avionics		79	79
Fuel System		10	10
Flight Controls		47	47
Hydraulics		14	14
Electrical		125	125
Furnishings		83	83
BEW Weight		1766	1623
POH Stated BEW		1620	1620
Battery Pack Weight Climb		101	149
Battery Pack Weight Cruise		0	53
Battery Pack Weight Descent		0	0
Fuel Weight		60	60
Fuel Volume (gal)		10	10
Weight (without cargo, crew/passengers)		1927	1859
Aircraft max gross		2535	2535
Available crew/passengers, cargo		**608**	**676**

Table 4-12: Breakdown of Flight Time and Fuel Consumption Serial-Hybrid DA40			
	Climb Phase	Cruise Phase	Descent Phase
Flight Time, min (hr)	9.4 (0.16)	60.8 (1.01)	23 (0.38)
Fuel Consumed, L (gal)	4.2 (1.1)	26.9 (7.1)	6.8 (1.8)
Fuel Consumption Rate, L/hr (gal/hr)	26.5 (7)	26.5 (7)	16 (4.25)
Generator Engine RPM [max rpm is 5800]	5800	5800	4600

The FAA requires that any GA aircraft land with 30 min of flight-time-at-cruise available as reserve and this corresponds to 13.2 L (3.5 gal) of fuel. The remaining fuel that is not consumed in the climb phase, during the descent phase, and saved for the reserve corresponds to 30.8 minutes. This means, the total flight time using the fixed engine system, is the sum of the climb phase (9.4 min), the cruise phase (30.8 min), and descent phase (23 min) for 63.3 minutes or 1.1 hours.

Table 4-13 compares the performance of all the DA40 variants discussed and allows more conclusions to be seen. The gas variant has the longest range with the ability to fly for 4.9 hours at cruise consuming fuel at around 30.3 L/hr (8 gal/hr). The serial-hybrid can fly for approximately 1 hour and the all-electric can fly for 1 hour at cruise. In the climb phase, the gas variant will consume 6.4 L (1.7 gal) of fuel while the serial-hybrid will consume 4.2 L (1.1 gal) of fuel and is a decrease of 35%. (The DA40 is already an efficient aircraft, which means at equivalent airspeeds, both aircraft consume similar amounts of fuel.)

The serial-hybrid variant has the flexibility to increase flight time by adding fuel tank volume, however, the tanks are limited to a volume that corresponds to approximately 2 hours of flight. After this limit, a single crew member cannot fit inside the aircraft. Figure 4-1 is a graph showing the relationship of fuel volume to aircraft weight without crew or cargo.

Table 4-13: DA40 Variant Performance Comparison				
	Gas	Electric	Baseline Serial-Hybrid	Fixed Serial-Hybrid
Cruise Altitude, m (ft)	2438 (8000)	2438 (8000)	2438 (8000)	2438 (8000)
CAS, m/s (kts)	56.6 (110)	56.6 (110)	56.6 (110)	56.6 (110)
EAS, m/s (kts)	56.5 (109.9)	56.5 (109.9)	56.5 (109.9)	56.5 (109.9)
TAS, m/s (kts)	63.8 (127)	63.8 (127)	63.8 (127)	63.8 (127)
Thrust, N	1113.9	1113.9	1113.9	1113.9
Power in by motor, kW (hp)	89 (119)	89 (119)	89 (119)	89 (119)
Time to Climb, min (hr)	9.3 (5.6)	9.3 (1)	9.3 (1.3)	9.3 (1.3)
Fuel Used Climb, L (gal)	6.4 (1.7)	0	4.2 (1.1)	4.2 (1.1)
Fuel Used Cruise, L (gal)	149.5 (39.5)	0	33.7 (8.9)	33.7 (8.9)
Discharge Rate “c-rate”	0.0	2.0	0	1.5

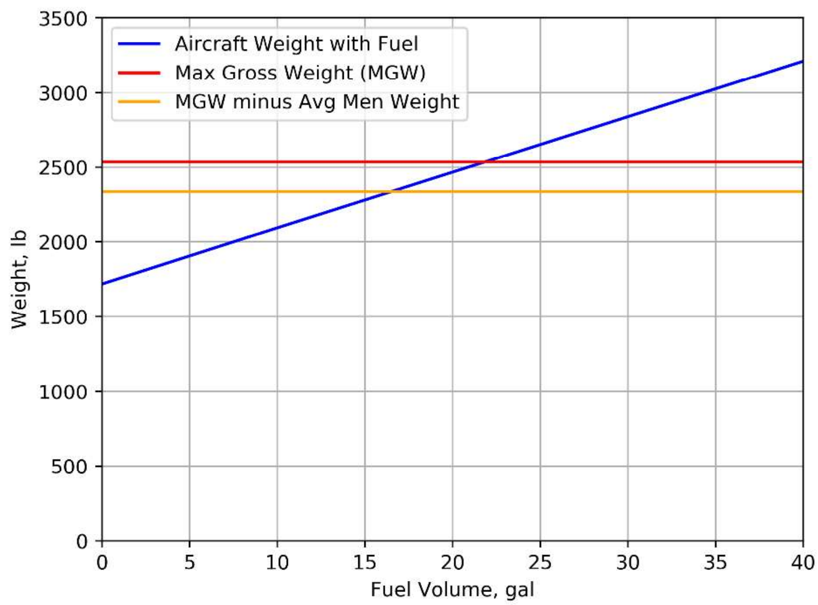

Figure 4-1: Fuel volume to weight relationship of a serial-hybrid system. The red line represents the max gross weight (MGW), and the orange line is the max gross weight minus the weight of a pilot. The blue line is the total aircraft weight without crew or cargo.

4.6 Serial-Hybrid Discussion

A serial-hybrid aircraft’s drawback is apparent when range and total flight time is compared to a traditional gas aircraft – it is unable to fly as far and for as long as the gas variant. It will use less fuel to achieve that flight time, but it is limited by the energy storage onboard since battery packs have a lower energy content per unit weight than gasoline. The feasibility of a serial-hybrid variation of an existing aircraft thus depends on the intended use case for the aircraft.

A use case for a hybrid aircraft is in short distance flights and in training environments where these flights are only for short periods and often in the immediate vicinity of the airport. A common flight for training aircraft is in an airport's traffic pattern (see Figure 2-1). The flight consists of takeoff, a steady climb to 304.8 m (1000 ft), flying parallel to the runway in the opposite direction of take-off, then descent, and landing. To climb to 304.8 m (1000 ft), Table 4-1 says 1 minute is needed to climb to 305 m (1000 ft) requiring a throttle setting of 134.2 kW (180 hp). Plugging that into Eq. 3-44, with a system efficiency 0.864, the total battery weight needed for the climb is 12.7 kg (28.0 lb). The power density of a lithium-polymer cell (listed in Table 2-7) is 217 Wh/kg, and means 2.75 kWh of capacity is needed. Battery pack capacity for a 2438 m (8000 ft) climb is 14.7 kWh, and so 5.3 take-offs to 305 m (1000 ft) could be done. Additional analysis is needed to determine the flight duration of a serial-hybrid aircraft flying an airport's traffic pattern as aircraft tend to fly slower overall or spend more time on the ground, and means there is less demand for energy from the generator engine. There is the possibility of recharging batteries since full power is not required from the generator engine.

Section 4.5 examined the flight time of a fixed engine-size DA40 following the mission profile in Figure 2-2. However, to better illustrate the trade-offs of using a serial-hybrid and the gas baseline, each aircraft will now perform a flight described in Figure 2-2 between two airports 556 km (300 NM) apart. First looking at the gas variant, the aircraft will take-off and climb from airport A, use 6.4 L (1.7 gal) and travel 18.5 km (9.9 NM). Descent from cruising altitude the aircraft will use 10.9 L (2.9 gal) of fuel and travel 84.2 km (45.5 NM). This leaves 416.0 km (224.6 NM) to travel in the cruise phase, which requires 1.93 hours to complete, and the aircraft will consume 71.5 L (18.9 gal). The total fuel consumption during the flight is 80.0 L (23.5 gal) and duration of the flight is 2.46 hours.

A serial-hybrid performing this flight will be different since intermediate stops between airport A and airport B are needed to recharge and refuel. In the climb phase it will consume 4.2 L (1.1 gal) of fuel and travel 18.5 km (9.9 NM). In the descent phase the aircraft will travel 84.2 km

(45.5 NM) and it will consume 6.8 L (1.8 gal) of fuel, which is assuming that the system is not charging the batteries. As with the gas-variant, 416.0 km (224.6 NM) are left to travel, but the serial-hybrid can only travel in the cruise phase for 30 minutes in order to land with 30 minutes flight time reserve. Traveling at 204 km/h (110 kts), the aircraft can travel 204 km (55 NM) and will consume 13.2 L (3.5 gal) of fuel. The total distance the aircraft traveled is 204.5 km (110.4 NM) and will have to stop at two interim airports before reaching the destination. The final cruise phase requires only 44.1 km (23.8 NM) of travel and uses 5.7 L (1.5 gal) of fuel. The aircraft will need to recharge on the ground and this can be achieved either using the generator engine or through an electrical connection. Assuming the aircraft generator recharges the batteries, an additional 15 L (4 gal) of fuel is needed. Total fuel usage, distance traveled, and Time En-Route is listed and totaled in Table 4-14. The serial-hybrid uses 10% less fuel than the gas-variant and requires 51% more time.

Table 4-14: Serial-Hybrid Totals from Airport A to Airport B.

		Fuel, L (gal)	Distance Traveled, km (NM)	Time En-Route, hr
Flight Phase	Climb 1	4.1 (1.1)	18.3 (9.9)	0.1
	Cruise 1	13.2 (3.5)	101.9 (55.0)	0.5
	Descent 1	6.8 (1.8)	84.3 (45.5)	0.4
	Charging	8.2 (2.0)	-	1.0
	Climb 2	4.1 (1.1)	18.3 (9.9)	0.1
	Cruise 2	13.2 (3.5)	101.5 (55.0)	0.5
	Descent 2	6.8 (1.8)	84.3 (45.5)	0.4
	Charging	8.2 (2.0)	-	1.0
	Climb 3	4.1 (1.1)	18.3 (9.9)	0.1
	Cruise 3	5.7 (1.5)	44.1 (23.8)	0.2
	Descent 3	6.8 (1.8)	84.3 (45.5)	0.4
Totals		80.3 (21.2)	556 (300)	4.80

4.7 Battery Recharging

Total travel time could be reduced if the generator engine operates with more power than is needed to sustain the descent portion of the flight. However, as the system is modeled the serial-hybrid system does not allow for inflight recharging during the cruise phase, and can only occur if the generator engine produces more power than needed by the main electric motor to sustain level flight. The C-rate describes the speed of discharge or charge of a battery pack where C-rate = 1 means the battery pack is discharged or charged in 1 hour. The total battery pack weight for the fixed power plant-weight DA40 is 91.6 kg (202 lb) and corresponds to a power capacity of 19.9 kWh. Since the system voltage is 400 V (dictated by the electric motor), the battery capacity can be expressed as 49.7 Ah as well. In order for the generator engine to sufficiently recharge the battery pack, it would need to supply 49.7 A or 19.9 kW to the battery in addition to the power to sustain straight and level flight. A larger engine means a larger BEW, which reduces the amount of crew, cargo, or energy storage, or a combination of the three.

Using ground charging stations for electric road vehicles as a guide, “fast” charging stations are quoting 24kW to 50kW of power supply [46]. A C-rate faster than 1 hour is possible, but battery life and thermal control become larger concerns at faster charge rates. However, the hybrid aircraft does have the ability to charge the battery pack while on the ground because the generator engine can operate and provide power. The generator engine only needs to supply 19.9kW (26.7 hp) and has the ability to provide 74.6kW (100 hp) at full power. 19.9 kW (26.7 hp) is on the low end of the engine’s power output which consumes approximately 7.6 L/hr (2 gal/hr). At least one hour is needed to recharge the batteries (at C-rate = 1) and will need at least 7.6 L (2 gal) of fuel. The airplane can then be refueled and is ready for the next flight.

In the descent phase, 10.9 kW of power is available to recharge the battery, and this would recharge the battery 4.2 kW or 21% of the battery capacity. (The descent phase is 23 minutes.) If the descent phase was used to recharge, then 10.2 L (2.7 gal) would be consumed, reducing the fuel available for cruise to 10.6 L (2.8 gal). Therefore, time sitting on the ground recharging can be reduced at the expense of time in the cruise phase.

5 Conclusions

Three different general aviation aircraft, a Diamond DA40, a Cessna Skyhawk 172S and a Cirrus SR22 were used to explore the possibility of converting the existing power plant system, a gas engine, to an all-electric or serial-hybrid power system. Feasibility was analyzed by comparing overall weight, useable weight, aircraft range and endurance, and the fuel economy of the serial-hybrid variant to the gas variant. The model developed for analysis provided weight characteristics and performance characteristics of the three aircraft. Weight breakdowns included individual component weights, power plant weights and battery pack weights. The power plant weight presented in the weight result tables used a statistical equation that is based on gasoline aircraft engines, and not serial-hybrid or all-electric aircraft, meaning the system weights reported should be viewed as estimates. Performance characteristics included system power, fuel consumption, flight time and distance.

Based on the results, the conversion of an existing aircraft to an all-electric would be difficult to successfully achieve and would require slower travel speeds and shorter flights. However, a serial-hybrid conversion is feasible with the main drawback being less flight endurance. At approximately equal amounts of available crew and cargo weight the gas-variant can fly for 4.9 hours at cruise, and the serial hybrid can fly for 1.3 hour at cruise, or approximately 1/4th as long. Long-distance trips would thus require more time and breaks than the existing gas variant a consequence of the higher energy storage density of hydrocarbon fuels compared to battery packs. However, unlike an all-electric aircraft, the serial-hybrid can recharge the onboard batteries without need for an electrical outlet. The serial-hybrid is viable in short distance flights or for use in specific scenarios such as pilot training, where a training flight would be able to perform at least 5 circuits in an airport's traffic pattern. In between these short flights, a serial-hybrid aircraft has time to recharge since the onboard engine and fuel provide sufficient power to change batteries in 1 hour or less. Fuel consumption is 10% less for traveling the same distance as the gas variant at the expense of total time needed to perform the trip. Less fuel is required for a short duration flight or a training flight as well.

6 References

[1] US EPA, O. Regulations for Greenhouse Gas Emissions from Aircraft. *US EPA*. https://www.epa.gov/regulations-emissions-vehicles-and-engines/regulations-greenhouse-gas-emissions-aircraft. Accessed Oct. 10, 2020.

[2] 2020 Toyota Prius Hybrid Electric Sedan | Be in Your Element. https://www.toyota.com/prius/. Accessed Jun. 10, 2020.

[3] Volt for Sale: 2019 Volt Pricing | Chevrolet. https://www.chevrolet.com/electric/volt-plug-in-hybrid/build-and-price/trim. Accessed Jun. 10, 2020.

[4] Brelje, B. J., and Martins, J. R. R. A. "Electric, Hybrid, and Turboelectric Fixed-Wing Aircraft: A Review of Concepts, Models, and Design Approaches." *Progress in Aerospace Sciences*, Vol. 104, 2019, pp. 1–19. https://doi.org/10.1016/j.paerosci.2018.06.004.

[5] Eviation's All-Electric Alice Regional Airliner Secures a Major Launch Order – AINtv. .

[6] 14 CFR Part 43 Subpart C - Maintenance, Preventive Maintenance, Rebuilding, and Alteration. In *Title 14: Aeronautics and Space*, 2016.

[7] 14 CFR Part 91 Subpart E - Maintenance, Preventive Maintenance, and Alterations. In *Title 14: Aeronautics and Space*, 2016.

[8] Vehicle Warranty. https://www.tesla.com/support/vehicle-warranty. Accessed Nov. 20, 2020.

[9] Model S. *Tesla*. https://www.tesla.com/models. Accessed Nov. 20, 2020.

[10] Harlow, J. E., Ma, X., Li, J., Logan, E., Liu, Y., Zhang, N., Ma, L., Glazier, S. L., Cormier, M. M. E., Genovese, M., Buteau, S., Cameron, A., Stark, J. E., and Dahn, J. R. "A Wide Range of Testing Results on an Excellent Lithium-Ion Cell Chemistry to Be Used as Benchmarks for New Battery Technologies." *Journal of The Electrochemical Society*, Vol. 166, No. 13, 2019, p. A3031. https://doi.org/10.1149/2.0981913jes.

[11] Kitty Hawk. https://kittyhawk.aero/. Accessed Jun. 10, 2020.

[12] Joby Aviation. https://www.jobyaviation.com/. Accessed Jun. 10, 2020.

[13] Wright Electric - Transforming the Aviation Industry with Electric Planes. *Wright Electric*. https://weflywright.com/. Accessed Jun. 10, 2020.

[14] Zunum Aero - Home Page Investment Opportunity. *Zunum Aero*. https://zunum.aero/. Accessed Mar. 29, 2020.

[15] E-Fan X - Electric Flight - Airbus. https://www.airbus.com/innovation/future-technology/electric-flight/e-fan-x.html. Accessed Mar. 29, 2020.

[16] Raymer, D. *Aircraft Design: A Conceptual Approach*. American Institute of Aeronautics and Astronautics, Playa del Rey, CA, 2018.

[17] EFlyer | Innovation for the Next Generation. https://electricflyer.com/. Accessed May 31, 2020.

[18] Eviation | Eviation Alice. https://www.eviation.co/. Accessed Jun. 10, 2020.

[19] AMPAIRE | Leading The Charge. https://www.ampaire.com/. Accessed May 31, 2020.

[20] VoltAero Innovation Provides the Power to Fly Safely. *VoltAero*. https://www.voltaero.aero/en/. Accessed May 31, 2020.

[21] ACCEL. https://www.rolls-royce.com/media/our-stories.aspx. Accessed May 31, 2020.

[22] Pipistrel ALPHA Electro - The First LSA Certified Electric Aircraft. Pipistrel USA. *Pipistrel-USA*. https://www.pipistrel-usa.com/alpha-electro/. Accessed Mar. 22, 2020.

[23] Pipistrel USA - About Our Company and It's History Starting in 1987. *Pipistrel-USA*. https://www.pipistrel-usa.com/about-us/. Accessed Mar. 22, 2020.

[24] Panthera. *Pipistrel-USA*. https://www.pipistrel-usa.com/panthera/. Accessed Mar. 22, 2020.

[25] *Aviation Maintenance Technician Handbook - Powerplant, Volume 1 (FAA-H-8083-32A)*. Federal Aviation Administration, Oklahoma City, OK, 2018.

[26] MagniX. *magniX*. https://magnix.aero/products/. Accessed Mar. 28, 2020.

[27] Airplane Flight Manual DA-40. Accessed Apr. 4, 2020.

[28] (L)IO-360-M1A Operation and Installation Manual. https://www.lycoming.com/sites/default/files/%28L%29IO-360-M1A%20Oper%20%26%20Install%20Manual%2060297-36.pdf.

[29] Information Manual Skyhawk SP Model 172S. Accessed Apr. 2, 2020.

[30] O-HO-IO-HIO-AIO & TIO-360 Oper Manual 60297-12.Pdf. https://www.lycoming.com/sites/default/files/O-HO-IO-HIO-AIO%20%26%20TIO-360%20Oper%20Manual%2060297-12.pdf. Accessed Apr. 2, 2020.

[31] *SR22 Airplane Flight Manual, P/N 13772-006*. Cirrus Design Corporation, 4515 Taylor Circle Duluth, MN 55811, 2020.

[32] Continental 500 Series AvGas Engine. *Continental Aerospace Technologies*. https://www.continentalmotors.aero/engines/500.aspx. Accessed Apr. 10, 2020.

[33] NCR18650BD | Panasonic Industrial Devices. https://na.industrial.panasonic.com/products/batteries/rechargeable-batteries/lineup/lithium-ion/series/90729/model/90731. Accessed Jun. 7, 2020.

[34] Jabiru Aircraft - Australian Made Recreational Aircraft Since 1988. *Jabiru*. https://jabiru.net.au/. Accessed Jun. 12, 2020.

[35] Rotax Aircraft Engines - PRODUCT RANGE - Rotax Aircraft Engines. https://www.flyrotax.com/products.html. Accessed Jun. 12, 2020.

[36] Gemini Aviation. https://www.geminidiesel.aero/aviation. Accessed Jun. 12, 2020.

[37] Welcome to Continental Motors, Inc Factory New and Rebuilt Aircraft Engines. http://www.continental.aero/. Accessed Jun. 12, 2020.

[38] Wilksch Aero. http://www.wilksch.aero/Home. Accessed Jun. 12, 2020.

[39] Pipistrel-Alpha-ELECTRO-Information-Pack.Pdf. http://52.203.47.21/wp-content/uploads/2018/03/Pipistrel-Alpha-ELECTRO-Information-Pack.pdf. Accessed Mar. 22, 2020.

[40] WORTMANN FX 63-137 AIRFOIL (Fx63137-Il). http://airfoiltools.com/airfoil/details?airfoil=fx63137-il#polars. Accessed Nov. 3, 2019.

[41] RONCZ LOW DRAG FLYING WING AIRFOIL (Marske7-Il). http://airfoiltools.com/airfoil/details?airfoil=marske7-il. Accessed Apr. 12, 2020.

[42] NACA 2412 (Naca2412-Il). http://airfoiltools.com/airfoil/details?airfoil=naca2412-il. Accessed Apr. 12, 2020.

[43] Airplane Flying Handbook. https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/airplane_handbook/media/airplane_flying_handbook.pdf. Accessed Mar. 18, 2020.

[44] FastStats. https://www.cdc.gov/nchs/fastats/body-measurements.htm. Accessed May 19, 2020.

[45] Rotax 912 ULS/S - Rotax Aircaft Engines. https://www.flyrotax.com/produkte/detail/rotax-912-uls-s.html. Accessed Apr. 8, 2020.

[46] Quick Guide to Fast Charging. *Charge Point*. https://www.chargepoint.com/files/Quick_Guide_to_Fast_Charging.pdf. Accessed May 31, 2020.

Appendix A – System Model Code

```
# Language: Python 3
# System calculator
# Imports bring in the variables and functions stored in other .py files.
# standard python libraries
import numpy as np
import math
import matplotlib.pyplot as plt

# variable and config files
from conversion_factors import *
from physical_constants import *
from aerodynamic_calcs_fxns import *
from air_properties import *
from WeightsEstimatesFxns import *
from yasa_electric_motor_prop import *
from seimens_electric_motor_prop import *
from battery_cell_prop import *

# # Plane Select
#
# The aircraft used in this analysis are Diamond Aircraft's DA-40, a Cessna 172S, and a Sirius SR22. These are all
#single engine aircraft that can be flown by pilots with a PPL. The DA40 and  C172 both are similar weights with
#similar amounts of rated engine horsepower. Their differences are in the difference in airframe design, material
#use, and other performance areas discussed later in the report. The C172 is a common plane used by flight
#schools for training pilots were a large majority flight activity is staying close to an airport, often just flying in a
#traffic pattern doing airport operations practice.
#
# The plane select variable changes which set of numbers are used throughout the calculation. These values were
#derived in a few different ways. The simplest was simply referring to the POH of the respective aircraft and using
#the listed value. The values were either directly listed to a simple calculation was needed to produce that value.
#For many of the surface areas, these were estimated based on the drawings provided in the POH. Dimensions
#were estimated by measuring the size of drawing and scaling those dimension using a scaling factor by measuring
#a known full size dimension. (The POH drawings all provided basic length, wing span, and height dimesons, which
#were used to determine that scaling factor.) The other method was using estimated values based primarily on the
#Aircraft Design Handbook and some literature.

Plane_Select = 0
Plane_List = ['DA40', 'C172', 'SR22']

if Plane_Select == 0:
    from DA40weights import *
    from DA40dim import *
    from DA40airfoil_prop_properties import *

if Plane_Select == 1:
    from C172weights import *
    from C172dim import *
    from C172airfoil_prop_properties import *

```

```
if Plane_Select == 2:
    from SR22weights import *
    from SR22dim import *
    from SR22airfoil_prop_properties import *

# # Configuration Selection
#
# Several configurations are considered to address both the accuracy of the model and potential feasibility of the
#model. The gas configuration is seeing how accurately the model predicts the properties of the existing aircraft. If
#the values are close, then the model is considered accurate and findings for other other variants are considered a
#decent estimation of performance. A major question is if it would be possible to do a "drop-in replacement" or
#conversion to an all-electric or hybrid configuration.
#
# Changing the Config_Select variable changes which combination of efficiencies are used.
#
# 0. Gas
# * This is using an ICE and is modeling the existing aircraft.
# 1. All Electric
# * The gas engine is removed and replaced with an electric engine. Battery cells are added as well.
# 2. Serial Hybrid
# * Similar to the electric engine variant, but a smaller gas engine is added to the system as well. Enough battery
#energy is added so that when the electric motor and gas engine are operating together, they produce sufficient
#power needed by the electric motor to climb up to the final cruising altitude.
# 3. No batteries/ turboelectric
# * The idea behind this is that a gas motor generates all the power needed and there are not batteries in the
#system.

Config_Select = 2

Config_List = ["Gas", "Electric", "Serial", "No Batteries Serial"]

Gas_Config = False
All_Electric_Config = False
Serial_Hybrid_Config = False
No_Batteries_Config = False #Serial hybrid without any batteries. Motors provice all required electricity.

if Config_Select == 0:
    Gas_Config = True
elif Config_Select == 1:
    All_Electric_Config = True
elif Config_Select == 2:
    Serial_Hybrid_Config = True
elif Config_Select == 3:
    No_Batteries_Config = True
elif Config_Select >= 4:
    print("Config_Select too large")

# # Weigh Estimation, Variables
Pick_Engines = True

cruise_alt = 1000 #ft
```

```
#electric motor hp/lb
array_of_densities_hplb = np.array([motor_power_cont_400_hp/weight_400_lbs,
motor_power_cont_750_hp/weight_750_lbs, motor_power_cont_SP70D_hp/weight_SP70D_lbs,
                        motor_power_cont_SP55D_hp/weight_SP55D_lbs,
motor_power_cont_SP260D_hp/weight_SP260D_lbs, motor_power_cont_SP200D_hp/weight_SP200D_lbs])
electric_motor_average_weight_lb = np.average(np.array([weight_400_lbs, weight_750_lbs, weight_SP70D_lbs,
weight_SP55D_lbs, weight_SP260D_lbs, weight_SP200D_lbs]))
average_den_hplb = np.average(array_of_densities_hplb)

array_of_densities = np.array([motor_power_cont_400/weight_400, motor_power_cont_750/weight_750,
motor_power_cont_SP70D/weight_SP70D,
                        motor_power_cont_SP55D/weight_SP55D, motor_power_cont_SP260D/weight_SP260D,
motor_power_cont_SP200D/weight_SP200D])
electric_motor_average_weight = np.average(np.array([weight_400, weight_750, weight_SP70D, weight_SP55D,
weight_SP260D, weight_SP200D]))
average_den = np.average(array_of_densities)

#light sport engine hp/lb
array_of_LS_weight = np.array([178, 103, 108, 134, 141, 167, 191, 170, 262, 280]);
array_of_LS_hp = np.array([120, 50, 65, 81, 100, 115, 100, 100, 100, 120]);
array_of_LS_density = array_of_LS_weight / array_of_LS_hp

average_LS_den = np.average(array_of_LS_density)
average_LS_weight = np.average(array_of_LS_weight)
average_LS_hp = np.average(array_of_LS_hp)

# # Power Requirements
### Initial Climb
# The initial climb of an aircraft is when the plane is departing the runway and trying to climb to its desired
#altitude or some intermediary altitude.


# This next section shows the needed power output for the plane at max gross weight listed in the POH.

def steady_angled_flight_fxn(Cfe, wet_ref_ratio, velocity, angle, Cla, aspect_ratio, sweep_angle, SoS, altitude,
airplane_weight, air_den_alt, disk_area, prop_effeciency, prop_rps, S):
    little_e = 1.78*(1-0.45*wet_ref_ratio**0.68)-0.64

    CL_alpha = CL_alpha_fxn(velocity, Cla, aspect_ratio, sweep_angle, SoS, altitude)
    K = K_fxn(aspect_ratio, CL_alpha, S)
    coeff_lift = coeff_lift_fxn(airplane_weight, climb_angle, air_den_alt, wing_ref_area, velocity)

    CDo = Cfe * wet_ref_ratio #estimation equation in text for parasite drag
    coeff_drag = CDo + K*(coeff_lift**2) #total drag coeffecient, combines parasite drag and induced drag. Induced
drag is a function of the coeffecient of lift.

    drag_force = drag_eqn_fxn(coeff_drag, air_den_alt, velocity, wing_ref_area)

    if angle >=0:
        thrust = drag_force + (airplane_weight * np.sin(np.abs(angle)))
    else:
        thrust = drag_force - (airplane_weight * np.sin(np.abs(angle)))
```

```
    v_vert = velocity * ((thrust - drag_force)/airplane_weight)

    power_out = (thrust * velocity)
    power_out_hp = (power_out/1000) / hp_to_kw
    power_in = (thrust * velocity)/prop_effeciency
    power_in_hp = (power_in/1000) / hp_to_kw
    advance_ratio = velocity/(prop_rps*prop_diam)

    return v_vert, power_out, power_in, power_out_hp, power_in_hp, thrust, drag_force, coeff_drag, CDo,
coeff_lift, CL_alpha, little_e, K, advance_ratio

def calc_v_speed(pressure, pressure_sea, air_den_alt, air_den_sea, velocity, CAS_kts, SoS):

    n=0
    converge = 0.1
    mach_num = CAS_kts/(SoS/kts_to_mps)
    while (converge)  > 0.0000001:

        if n > 0:
            mach_num  = mach_num1

        qc = pressure*(((1+0.2*(mach_num)**2)**3.5) - 1)
        EAS_num = (((qc/pressure)+1)**(2/7)) - 1
        EAS_denom = (((qc/pressure_sea)+1)**(2/7)) - 1
        CAS = CAS_kts
        EAS = CAS * np.sqrt(pressure/pressure_sea) * (EAS_num/EAS_denom)**0.5
        TAS = EAS / np.sqrt(air_den_alt/air_den_sea)

        mach_num1 = TAS/(SoS/kts_to_mps)

        converge = np.abs(1 - np.abs(mach_num/mach_num1))

        #velocity_difference = np.abs(velocity_out - velocity_in)
        if n == 1000:
            print('break')

            velocity_kts = TAS
            velocity = velocity_kts * kts_to_mps
            break
        else:
            n = n + 1


    velocity_kts = TAS
    velocity = velocity_kts * kts_to_mps
    return velocity, velocity_kts, EAS, TAS

#propeller force requirements
iteration = np.argmax(alt_air == cruise_alt) + 1
#define storage arrays for use in keeping variables and writing to excel file.
power_in_climb_store = np.zeros(iteration-1)
power_out_climb_store = np.zeros(iteration-1)
v_vert_climb_store = np.zeros(iteration-1)
```

```
climb_angle_store = np.zeros(iteration-1)
time_segment_climb = np.zeros(iteration-1)
TAS_climb_store = np.zeros(iteration-1)
EAS_climb_store = np.zeros(iteration-1)
CAS_climb_store = np.zeros(iteration-1)
e_climb_store = np.zeros(iteration-1)
K_climb_store = np.zeros(iteration-1)
CDo_climb_store = np.zeros(iteration-1)
coeff_lift_climb_store = np.zeros(iteration-1)
coeff_drag_climb_store = np.zeros(iteration-1)
drag_force_climb_store = np.zeros(iteration-1)
dyn_press_climb_store = np.zeros(iteration-1)
CL_alpha_climb_store = np.zeros(iteration-1)
advance_ratio_climb_store = np.zeros(iteration-1)
thrust_climb_store = np.zeros(iteration-1)

climb_angle_deg = 20 #arbitrary
climb_angle = climb_angle_deg * (np.pi / 180)

power_in_climb_max = engine_power * 1000

prop_rpm = 2500
prop_rps = prop_rpm/60

prop_effeciency = 0.8

S_wing = S

CAS_climb = Vy_CAS

v_climb_kts = CAS_climb
v_climb = v_climb_kts * kts_to_mps #convert to m/s

c = 0
for c in range(1, iteration):

    Cfe_climb = 0.0055 #constant from text

    altitude = alt_air[c] #ft
    entry = np.argmax(alt_air == altitude)
    air_den_alt = density_air[entry]
    air_den_sea = density_air[1]
    SoS = speed_of_sound[entry]
    pressure = absolute_press_air[entry] * 10000
    pressure_sea = absolute_press_air[1] * 10000

    [v_climb, v_climb_kts, EAS_climb, TAS_climb] = calc_v_speed(pressure, pressure_sea, air_den_alt, air_den_sea,
v_climb, CAS_climb, SoS)

    dyn_press_climb = dynamic_pressure_fxn(air_den_alt, v_climb)

    [v_vert_climb, power_out_climb, power_in_climb, power_out_hp_climb, power_in_hp_climb,
```

```
     thrust_climb, drag_force_climb, coeff_drag_climb, CDo_climb, coeff_lift_climb, CL_alpha_climb, little_e_climb,
K_climb, advance_ratio_climb] = steady_angled_flight_fxn(Cfe_climb, wet_ref_ratio, v_climb, climb_angle, Cla_10,
aspect_ratio, sweep_angle, SoS, altitude, airplane_weight, air_den_alt, disk_area, prop_effeciency, prop_rps,
S_wing)

    [v_exhaust_climb_act, power_out_climb_act, power_in_climb_act, power_out_hp_climb_act,
power_in_hp_climb_act, power_out_check_climb_act, effeciency_climb_act] =
actuator_disk_fxn(drag_force_climb, airplane_weight, thrust_climb, disk_area, air_den_alt, v_climb)

    while power_in_climb > power_in_climb_max:
        climb_angle = climb_angle - 0.0001
        [v_vert_climb, power_out_climb, power_in_climb, power_out_hp_climb, power_in_hp_climb,
         thrust_climb, drag_force_climb, coeff_drag_climb, CDo_climb, coeff_lift_climb, CL_alpha_climb,
little_e_climb, K_climb, advance_ratio_climb] = steady_angled_flight_fxn(Cfe_climb, wet_ref_ratio, v_climb,
climb_angle, Cla_10, aspect_ratio, sweep_angle, SoS, altitude, airplane_weight, air_den_alt, disk_area,
prop_effeciency, prop_rps, S_wing)

        [v_exhaust_climb_act, power_out_climb_act, power_in_climb_act, power_out_hp_climb_act,
power_in_hp_climb_act, power_out_check_climb_act, effeciency_climb_act] =
actuator_disk_fxn(drag_force_climb, airplane_weight, thrust_climb, disk_area, air_den_alt, v_climb)

        if climb_angle < (0.01*(np.pi/180)):
            print("break")
            break

    v_vert_climb_fpm = v_vert_climb * mps_to_fpm

    power_in_climb_store[c-1] = power_in_climb
    power_out_climb_store[c-1] = power_out_climb
    v_vert_climb_store[c-1] = v_vert_climb
    climb_angle_store[c-1] = climb_angle
    time_segment_climb[c-1] = 1000 / v_vert_climb_fpm
    TAS_climb_store[c-1] = TAS_climb
    EAS_climb_store[c-1] = EAS_climb
    CAS_climb_store[c-1] = CAS_climb
    e_climb_store[c-1] = little_e_climb
    K_climb_store[c-1] = K_climb
    CDo_climb_store[c-1] = CDo_climb
    coeff_lift_climb_store[c-1] = coeff_lift_climb
    coeff_drag_climb_store[c-1] = coeff_drag_climb
    drag_force_climb_store[c-1] = drag_force_climb
    dyn_press_climb_store[c-1] = dyn_press_climb
    CL_alpha_climb_store[c-1] = CL_alpha_climb
    advance_ratio_climb_store[c-1] = advance_ratio_climb
    thrust_climb_store[c-1] = thrust_climb

#plots of climb rate over alt

fig1 = plt.figure()
ax1 = fig1.add_subplot(111)
ax1.plot(alt_air[1:iteration], (v_vert_climb_store * mps_to_fpm))
ax1.set_xlabel("Altitude, ft")
ax1.set_ylabel("Climb Rate, fpm")
```

```
ax1.set_title(Plane_List[Plane_Select] + " Climb Rate")
yaxis_top = max(v_vert_climb_store * mps_to_fpm) + max(v_vert_climb_store * mps_to_fpm)*0.1
ax1.set_ylim(0, yaxis_top)

fig1.savefig(("outputs" + "\\" + Plane_List[Plane_Select] + " climb_rate" + ".png" ), dpi=400, transparent=False)

####################################################
#propeller force requirements during cruise
#####################################################

Cfe_cruise = 0.0055 #constant from text
little_e_cuise = 0.8 #constant from text

#cruise_alt= 1000 #ft
entry = np.argmax(alt_air == cruise_alt)
air_den_alt = density_air[entry]
air_den_sea = density_air[1]
SoS = speed_of_sound[entry]
pressure = absolute_press_air[entry] * 10000
pressure_sea = absolute_press_air[1] * 10000

#CAS_cruise = 90
CAS_cruise = v_cruise_CAS
v_cruise_kts = CAS_cruise
v_cruise = v_cruise_kts * kts_to_mps #convert to m/s
[v_cruise, v_cruise_kts, EAS_cruise, TAS_cruise] = calc_v_speed(pressure, pressure_sea, air_den_alt, air_den_sea,
v_cruise, CAS_cruise, SoS)

Cla_cruise = Cla_10
CL_alpha_cruise = CL_alpha_fxn(v_cruise, Cla_cruise, aspect_ratio, sweep_angle, SoS, cruise_alt)

K_cruise = K_fxn(aspect_ratio, CL_alpha_cruise, S)

#dyn_press_cruise = dynamic_pressure_fxn(air_den_sea_level, v_cruise)
dyn_press_cruise = dynamic_pressure_fxn(air_den_alt, v_cruise)

cruise_angle = 0
effeciency_cruise = 0.9

CDo_cruise = Cfe_cruise * wet_ref_ratio #estimation equation in text for parasite drag

[v_vert_cruise, power_out_cruise, power_in_cruise, power_out_hp_cruise, power_in_hp_cruise, thrust_cruise,
drag_force_cruise, coeff_drag_cruise, CDo_cruise, coeff_lift_cruise, CL_alpha_cruise, little_e_cruise, K_cruise,
advance_ratio_cruise] = steady_angled_flight_fxn(Cfe_cruise, wet_ref_ratio, v_cruise, cruise_angle, Cla_cruise,
aspect_ratio, sweep_angle, SoS, cruise_alt, airplane_weight, air_den_alt, disk_area, prop_effeciency, prop_rps, S)

v_exhaust_cruise = 1

#cruise_alt = 8000 #ft
iteration = np.argmax(alt_air == cruise_alt) + 1

power_in_descent_store = np.zeros(iteration-1)
power_out_descent_store = np.zeros(iteration-1)
```

```
v_vert_descent_store = np.zeros(iteration-1)
descent_angle_store = np.zeros(iteration-1)
time_segment_descent = np.zeros(iteration-1)
TAS_descent_store = np.zeros(iteration-1)
EAS_descent_store = np.zeros(iteration-1)
CAS_descent_store = np.zeros(iteration-1)
e_descent_store = np.zeros(iteration-1)
K_descent_store = np.zeros(iteration-1)
CDo_descent_store = np.zeros(iteration-1)
coeff_lift_descent_store = np.zeros(iteration-1)
coeff_drag_descent_store = np.zeros(iteration-1)
drag_force_descent_store = np.zeros(iteration-1)
dyn_press_descent_store = np.zeros(iteration-1)
CL_alpha_descent_store = np.zeros(iteration-1)
advance_ratio_descent_store = np.zeros(iteration-1)
thrust_descent_store = np.zeros(iteration-1)

power_in_max = engine_power * 1000

prop_rpm = 2500
prop_rps = prop_rpm/60

prop_effeciency = 0.8

S_wing = S

v_vert_descent_fpm = -350 #fpm
v_vert_descent = v_vert_descent_fpm / mps_to_fpm

CAS_descent = v_cruise_CAS
v_descent_kts = CAS_descent
v_descent = v_descent_kts * kts_to_mps

descent_angle = np.arcsin(v_vert_descent/v_descent)

Cfe_descent = 0.0055 #constant from text
little_e_descent = 0.8 #constant from text

c = 0
for c in range(1, iteration):

    altitude = alt_air[c] #ft
    entry = np.argmax(alt_air == altitude)
    air_den_alt = density_air[entry]
    air_den_sea = density_air[1]
    SoS = speed_of_sound[entry]
    pressure = absolute_press_air[entry] * 10000
    pressure_sea = absolute_press_air[1] * 10000

    [v_descent, v_descent_kts, EAS_descent, TAS_descent] = calc_v_speed(pressure, pressure_sea, air_den_alt,
air_den_sea, v_descent, CAS_descent, SoS)

    dyn_press_descent = dynamic_pressure_fxn(air_den_alt, v_descent)
```

```

    descent_angle = np.pi/2 - np.arccos(v_vert_descent/v_descent)

    Cla_descent = Cla_10

    [v_vert_descent, power_out_descent, power_in_descent, power_out_hp_descent, power_in_hp_descent,
thrust_descent, drag_force_descent, coeff_drag_descent, CDo_descent, coeff_lift_descent, CL_alpha_descent,
little_e_descent, K_descent, advance_ratio_descent] = steady_angled_flight_fxn(Cfe_descent, wet_ref_ratio,
v_descent, descent_angle, Cla_descent, aspect_ratio, sweep_angle, SoS, cruise_alt, airplane_weight, air_den_alt,
disk_area, prop_effeciency, prop_rps, S_wing)

    v_vert_descent_fpm = v_vert_descent * mps_to_fpm

    power_in_descent_store[c-1] = power_in_descent
    power_out_descent_store[c-1] = power_out_descent
    v_vert_descent_store[c-1] = v_vert_descent
    descent_angle_store[c-1] = descent_angle
    time_segment_descent[c-1] = 1000 / v_vert_descent_fpm
    TAS_descent_store[c-1] = TAS_descent
    EAS_descent_store[c-1] = EAS_descent
    CAS_descent_store[c-1] = CAS_descent
    e_descent_store[c-1] = little_e_descent
    K_descent_store[c-1] = K_descent
    CDo_descent_store[c-1] = CDo_descent
    coeff_lift_descent_store[c-1] = coeff_lift_descent
    coeff_drag_descent_store[c-1] = coeff_drag_descent
    drag_force_descent_store[c-1] = drag_force_descent
    dyn_press_descent_store[c-1] = dyn_press_descent
    CL_alpha_descent_store[c-1] = CL_alpha_descent
    advance_ratio_descent_store[c-1] = advance_ratio_descent
    thrust_descent_store[c-1] = thrust_descent

#plots of climb rate over alt

fig2 = plt.figure()
ax2 = fig2.add_subplot(111)
ax2.plot(alt_air[1:iteration], (v_vert_descent_store * mps_to_fpm))
ax2.set_xlabel("Altitude, ft")
ax2.set_ylabel("Descent Rate, fpm")
ax2.set_title(Plane_List[Plane_Select] + " Descent Rate")
yaxis_top = max(v_vert_descent_store * mps_to_fpm) + max(v_vert_descent_store * mps_to_fpm)*0.1
ax2.set_ylim(0, yaxis_top)

fig2.savefig(("outputs" + "\\" + Plane_List[Plane_Select] + " descent_rate" + ".png" ), dpi=400, transparent=False)

# # Effeciencies
# Based on system sketch, different combination of effeciencies are use. 1 indicates that the portion does not exist

#effeciencies
if Gas_Config == True:
    fuel_to_motor = 0.3
    power_generation = 1
    gen_bus_control_em = 1
```

```
        battery_bus_control_em = 1
        em_effeciency = 1 #effeciency of electric motor from input electricity to output force
        output_shaft_belt_prop = 1
        prop_effeciency_climb = prop_effeciency
        prop_effeciency_cruise = effeciency_cruise

    if All_Electric_Config == True:
        fuel_to_motor = 1
        power_generation = 1
        gen_bus_control_em = 1
        battery_bus_control_em = 0.95
        em_effeciency = 0.96 #effeciency of electric motor from input electricity to output force
        output_shaft_belt_prop = 1
        prop_effeciency_climb = prop_effeciency
        prop_effeciency_cruise = effeciency_cruise

    if Serial_Hybrid_Config == True:
        fuel_to_motor = 0.3 #1
        power_generation = 0.95 #2
        gen_bus_control_em = 0.9  #3, 5, 6
        battery_bus_control_em = 0.9 #4
        em_effeciency = 0.96 #7 -- effeciency of electric motor from input electricity to output force to prop shaft
        output_shaft_belt_prop = 1 #this is covered by the propeller effeciency
        prop_effeciency_climb = prop_effeciency
        prop_effeciency_cruise = effeciency_cruise

    if No_Batteries_Config == True:
        fuel_to_motor = 0.3
        power_generation = 0.95
        gen_bus_control_em = 0.95
        battery_bus_control_em = 1
        em_effeciency = 0.96 #effeciency of electric motor from input electricity to output force
        output_shaft_belt_prop = 1
        prop_effeciency_climb = prop_effeciency
        prop_effeciency_cruise = effeciency_cruise

    # # Hybrid Motor Mass
    # This describes the entire hybrid motor system. Which is a gas generator, electric motor to act as electric
    generator and attached to the gas generator, then the motor used to drive the propeller.
    # * Weight for the gas generator is based of several light-sport engines.
    # * Power requirements are sized on cruise power.

    #hybrid motor mass
    needed_power_hp = power_in_hp_cruise

    if Pick_Engines == True:
        if Plane_Select == 0 or Plane_Select == 1:
            hp_from_engine_gen = 100 #ROTAX 912 S/ULS
            hybrid_gas_weight_lb = 141
            hybrid_gen_weight_lb = 53 #YASA 400
            hybrid_em_weight_lb = 108 #SP200D
            gal_per_hr = 5.5
            em_voltage = 400
```

```
    if Plane_Select == 2:
        hp_from_engine_gen = 180 #power of a da40 or C172
        hybrid_gas_weight_lb = 300
        hybrid_gen_weight_lb = 108 #YASA 400
        hybrid_em_weight_lb = 110 #SP260D
        gal_per_hr = 12
        em_voltage = 400

    if power_in_cruise/1000 > (hp_from_engine_gen * hp_to_kw):
        power_from_gen = (hp_from_engine_gen * hp_to_kw)*1000
    else:
        power_from_gen = power_in_cruise

else:
    hp_from_engine_gen = needed_power_hp / (gen_bus_control_em * em_effeciency * output_shaft_belt_prop *
power_generation)
    hybrid_gas_weight_lb = hp_from_engine_gen * average_LS_den
    hybrid_gen_weight_lb = needed_power_hp/average_den_hplb
    hybrid_em_weight_lb = power_in_hp_climb/average_den_hplb
    power_from_gen = power_in_cruise
    em_voltage = 400
    gal_per_hr = 7

hybrid_motor1_weight_lb = hybrid_gas_weight_lb + hybrid_gen_weight_lb + hybrid_em_weight_lb

# # Weight Estimates
# The next two sections determine estimates of weights based on statistical equations for typical planes.
#
# This is the list of variables used in estimating weights.
#
# The variables that change in configuration is
# * W_en -- engine weight
# * N_t -- number of fuel tanks
# * V_t -- total fuel volume
# * N_en -- number of engines in aircraft
# * W_fw -- This number changes based on V_t and is simply gallons of fuel times 6.
#     * It will change depending on the configuration. All electric converts the volume to be battery volume times
#density

velocity = v_cruise_CAS * kts_to_mps

#variables
A = aspect_ratio #aspect ratio
B_w = wing_span_ft #wing span, ft

D = height_ft #fuselage structural depth

K_h = 0.05 #0.05 for low subsonic with hydraulics for brakes and retracts only

L = overall_length_ft #fuselage strectural length, ft
L_m = 0.588 / in_to_m  #extended length of main landing gear, in
L_t = 4.641 * ft_to_m  #tail length; wing qwuarter-MAC to tail quarter-MAC, ft
```

```
M = mach_number_NE #mach number (design maximum)

P_delta = 8 #cabin pressure differential, typically 8psi

S_f = fuselage_area_ft #fuselage area, sq.ft
S_ht = plane_area_ft[0] #horozontal tail area, sq.ft
S_vt = plane_area_ft[1] #vertial tail area, sq.ft
S_w = wing_area_ft #trapezoidal wing area, sq.ft

V_pr = 0 #volume of pressurized section

if Gas_Config == True:
    W_en = engine_weight_lb #engine weight, each lb
    N_t = 2 #number of fuel tanks
    V_t = total_fuel #total fuel volume, gal
    W_fw = V_t * 6 #weight of fuel in wing, if zero ignore, lb

elif All_Electric_Config == True:
    W_en = electric_motor_average_weight_lb #lb
    N_t = 0 #number of fuel tanks
    V_t = 0 #41.2 #total fuel volume, gal
    W_fw = 0 #V_t * cell_mass_density #weight of fuel in wing, if zero ignore, lb

elif Serial_Hybrid_Config == True:
    W_en = hybrid_motor1_weight_lb
    N_t = 2 #number of fuel tanks
    V_t = 10 #total fuel volume, gal
    W_fw = V_t * 6 #weight of fuel in wing, if zero ignore, lb
    V_t_loop = np.arange(0, int(total_fuel), 1)

elif No_Batteries_Config == True:
    W_en = hybrid_motor2_weight_lb
    N_t = 2 #number of fuel tanks
    V_t = total_fuel #total fuel volume, gal
    W_fw = V_t * 6 #weight of fuel in wing, if zero ignore, lb

#W_en = lycoming_weight_lb #engine weight, each lb

W_dg = airplane_mass_lbs # flight design gross weight, lb
W_l = airplane_mass_lbs #landing desigh gross weight, lb
W_press = 11.9*((V_pr*P_delta)**0.271) #weight penalty due to pressurization.
W_uav = 0.03 * BEW_lbs #unistalled avionics weight, lb (typically = 800 to 1400lb) #see table 11.6

H_t = 0.392 / 0.0254 #horozontal tail height above fuselage
H_v = 0.392 / 0.0254 #vertial tail height above fuselage

if Plane_Select == 0:
    H_t_H_v = 1 #0 fior conventional tail, 1.0 for T tail
    V_i = total_fuel/2 #integral tanks volume, gal
elif Plane_Select == 1:
    H_t_H_v = 0 #0 fior conventional tail, 1.0 for T tail
    V_i = total_fuel/2 #integral tanks volume, gal
```

```
elif Plane_Select == 2:
    H_t_H_v = 0 #0 fior conventional tail, 1.0 for T tail
    V_i = total_fuel/2 #integral tanks volume, gal

sweep = sweep_angle
sweep_ht = 0
sweep_vt = 0

q = dynamic_pressure_fxn(air_den_sea_level_slug * slugs_to_lb, velocity) #dynamic pressure at cruise
lmbda = 0.61 #taper ratio
lmbda_h = 0.57 #taper ratio for tail
lmbda_vt = 0.82 #taper ratio for vert tail
thick_to_chord = thickness_in/MAC_in #0.44 #thickness to chord ratio, use average

N_en = 1 #number of engines (total for aircraft)
N_l = 3 * 1.5 #ultimate landing load factor. = N_gear x 1.5
N_p = 4 # number of personnel onboard (crew and passengers)

N_z = 4.4 * 1.5 #ultimate load factor, = 1.5 x limit load factor see pg 494/495

#eqns
if Plane_Select == 0 or Plane_Select == 2:
    W_wing = W_wing_fxn(S_w, W_fw, A, sweep, q, lmbda, thick_to_chord, N_z, W_dg) * 0.9

    W_horo_tail = W_horo_tail_fxn(N_z, W_dg, q, S_ht, thick_to_chord, sweep, A, sweep_ht, lmbda_h) * 0.88

    W_vert_tail = W_vert_tail_fxn(H_t_H_v, N_z, W_dg, q, S_vt, thick_to_chord, A, sweep_vt, lmbda_vt) * 0.88

    W_fuselage = W_fuselage_fxn(S_f, N_z, W_dg, L_t, L, D, q, W_press) * 0.95

else:
    W_wing = W_wing_fxn(S_w, W_fw, A, sweep, q, lmbda, thick_to_chord, N_z, W_dg)

    W_horo_tail = W_horo_tail_fxn(N_z, W_dg, q, S_ht, thick_to_chord, sweep, A, sweep_ht, lmbda_h)

    W_vert_tail = W_vert_tail_fxn(H_t_H_v, N_z, W_dg, q, S_vt, thick_to_chord, A, sweep_vt, lmbda_vt)

    W_fuselage = W_fuselage_fxn(S_f, N_z, W_dg, L_t, L, D, q, W_press)

[W_landing_gear, W_main_landing_gear, W_nose_landing_gear] = W_landing_gear_fxn(N_l, W_l, L_m)

W_installed_engine_total = W_installed_engine_total_fxn(W_en, N_en)

W_fuel_system = W_fuel_system_fxn(V_t, V_i, N_t, N_en)

W_flight_controls = W_flight_controls_fxn(L, B_w, N_z, W_dg)

W_hydraulics = W_hydraulics_fxn(K_h, W_dg, M)

W_avionics = W_avionics_fxn(W_uav)

W_electrical = W_electrical_fxn(W_fuel_system, W_avionics)
```

```
W_air_con_and_anti_ice = W_air_con_and_anti_ice_fxn(W_dg, N_p, W_avionics, M)

W_furnishings = W_furnishings_fxn(W_dg)

W_array = np.array([W_wing, W_horo_tail, W_vert_tail, W_fuselage, W_main_landing_gear,
W_nose_landing_gear, W_installed_engine_total, W_avionics, W_fuel_system, W_flight_controls, W_hydraulics,
W_electrical, W_furnishings], dtype = "float64")

W_sum = np.sum(W_array)

# # Battery Mass Fractions
# Battery mass is determined for climb and cruise portions of the mission. These equations are based on some of
#the previous aerodynamic assumptions used to determine power requirements of the propeller.
# Battery mass for all electric

def battery_mass_known_run_time_fxn(time_to_run_hr, power_used, cell_density, effeciency):
    batt_mass = (1000 * time_to_run_hr * power_used)/(cell_density * effeciency)
    return batt_mass

#climb battery weight
time_to_climb_hr = np.sum(time_segment_climb) / 60

effeciency = (battery_bus_control_em * em_effeciency)

if Serial_Hybrid_Config == True:
    #determine climb power average
    climb_power_average = (np.average(power_in_climb_store) - power_from_gen)/1000

    W_batt_climb = battery_mass_known_run_time_fxn(time_to_climb_hr, climb_power_average,
cell_grav_density, effeciency)
    W_batt_climb = W_batt_climb + W_batt_climb * 0.2
    W_batt_climb_lb = W_batt_climb * lb_to_kg
    W_batt_frac_climb = W_batt_climb / airplane_mass
    #determine cruise time of cruise
    fuel_used_climb = np.sum(time_segment_climb/60) * gal_per_hr
    time_cruise_hr = (V_t - fuel_used_climb) / gal_per_hr
    print(time_cruise_hr)
    wing_loading = airplane_mass / wing_area
    range_cruise_nm = v_cruise * time_cruise_hr

    cruise_power_average = (power_in_cruise-power_from_gen)/1000
    if cruise_power_average == 0:
        W_batt_cruise = 0
    else:
        W_batt_cruise = battery_mass_known_run_time_fxn(time_cruise_hr, cruise_power_average ,
cell_grav_density, effeciency)
    W_batt_cruise_lb = W_batt_cruise * lb_to_kg
    W_batt_frac_cruise = W_batt_cruise / airplane_mass

    #desired charge and discharge rates for battery packs
    c_rate_charge = 0.3
    c_rate_dis = 0.3
```

```
    #determine the amp discharge rate and the total amp-hous of the batters
    print(W_batt_cruise)
    battery_pack_capacity_cruise = (W_batt_cruise / lb_to_kg) * cell_grav_density
    batt_amps_draw_cruise = (cruise_power_average * 1000) / em_voltage
    batt_amps_capacity_cruise =  battery_pack_capacity_cruise / em_voltage
    print(batt_amps_capacity_cruise)

    #determine discharge of battery pack
    if batt_amps_draw_cruise == 0:
        actual_discharge_rate = 0
    else:
        actual_discharge_rate = batt_amps_draw_cruise / batt_amps_capacity_cruise

    print(actual_discharge_rate)
    motor_gen = power_from_gen
    batt_pack_volts = em_voltage

    descent_power_average = -(np.average(power_in_descent_store)/1000) - (power_from_gen/1000)
    time_to_descent_hr = np.sum(time_segment_descent) / 60


elif All_Electric_Config == True:
    climb_power_average = np.average(power_in_climb_store)/1000

    W_batt_climb = battery_mass_known_run_time_fxn(time_to_climb_hr, climb_power_average,
cell_grav_density, effeciency)
    W_batt_climb = W_batt_climb + W_batt_climb * 0.2
    W_batt_climb_lb = W_batt_climb * lb_to_kg
    W_batt_frac_climb = W_batt_climb / airplane_mass

    #cruise battery requirements
    time_cruise_hr = 1
    fuel_used_climb = 0
    wing_loading = airplane_mass / wing_area
    range_cruise_nm = v_cruise * time_cruise_hr

    cruise_power_average = (power_in_cruise)/1000
    W_batt_cruise = battery_mass_known_run_time_fxn(time_cruise_hr, cruise_power_average ,
cell_grav_density, effeciency)
    W_batt_cruise_lb = W_batt_cruise * lb_to_kg
    W_batt_frac_cruise = W_batt_cruise / airplane_mass
    #desired charge and discharge rates for battery packs
    c_rate_charge = 0.3
    c_rate_dis = 0.3

    #determine the amp discharge rate and the total amp-hours of the batters
    print(W_batt_cruise)
    W_batt_cruise = battery_mass_known_run_time_fxn(time_cruise_hr, power_in_cruise/1000, cell_grav_density,
effeciency)
    W_batt_cruise_lb = W_batt_cruise * lb_to_kg
    W_batt_frac_cruise = W_batt_cruise / airplane_mass

    #determine the amp discharge rate and the total amp-hous of the batters
```

```
    battery_pack_capacity_cruise = (W_batt_cruise / lb_to_kg) * cell_grav_density
    batt_amps_draw_cruise = (cruise_power_average * 1000) / em_voltage
    batt_amps_capacity_cruise =  battery_pack_capacity_cruise / em_voltage
    print(batt_amps_capacity_cruise)

    #determine discharge of battery pack
    actual_discharge_rate = batt_amps_draw_cruise / batt_amps_capacity_cruise
    print(actual_discharge_rate)
    motor_gen = power_from_gen
    batt_pack_volts = em_voltage

    descent_power_average = (np.average(power_in_descent_store)/1000)
    time_to_descent_hr = np.sum(time_segment_descent) / 60

else:
    climb_power_average = np.average(power_in_climb_store)/1000
    fuel_used_climb = np.sum(time_segment_climb/60) * climb_fuel_consumption
    time_cruise_hr = (V_t-fuel_used_climb) / cruise_fuel_consumption
    print(time_cruise_hr)
    wing_loading = airplane_mass / wing_area
    range_cruise_nm = v_cruise * time_cruise_hr

    W_batt_climb = 0
    W_batt_climb = W_batt_climb + W_batt_climb * 0.2
    W_batt_climb_lb = W_batt_climb * lb_to_kg
    W_batt_frac_climb = W_batt_climb / airplane_mass

    W_batt_cruise = 0
    W_batt_cruise_lb = W_batt_cruise * lb_to_kg
    W_batt_frac_cruise = W_batt_cruise / airplane_mass

    batt_amps_draw_cruise = 0
    batt_amps_capacity_cruise =  0

    actual_discharge_rate = 0
    motor_gen = power_from_gen
    batt_pack_volts = em_voltage

    W_batt_descent = 0
    W_batt_descent_lb = 0

#cruise battery weight

if Serial_Hybrid_Config == True or All_Electric_Config == True:
    battery_pack_capacity_climb = W_batt_climb * cell_grav_density
    climb_discharge_rate = 1000 * climb_power_average / battery_pack_capacity_climb

    battery_pack_capacity_cruise = (W_batt_cruise * cell_grav_density)
    if battery_pack_capacity_cruise == 0:
        cruise_discharge_rate = 0
    else:
        cruise_discharge_rate = power_in_cruise / battery_pack_capacity_cruise
```

```
    W_batt_descent = battery_mass_known_run_time_fxn(time_to_descent_hr, descent_power_average, cell_grav_density, effeciency)
    W_batt_descent_lb = W_batt_descent * lb_to_kg
    W_batt_frac_descent = W_batt_descent / airplane_mass


#mass Totals
if Gas_Config == True:
    total_mass = W_sum + W_fw

if All_Electric_Config == True:
    total_mass = W_sum + W_batt_climb_lb + W_batt_cruise_lb

if Serial_Hybrid_Config == True:
    if Pick_Engines == True:
        total_mass = W_sum + W_batt_climb_lb + W_batt_cruise_lb + W_fw
    else:
        total_mass = W_sum + W_batt_climb_lb + W_fw

if No_Batteries_Config == True:
    total_mass = W_sum + W_fw

#########################################
######fuel system analysis############
#########################################

if Serial_Hybrid_Config == True:

    W_fuel_system_loop = np.zeros(len(V_t_loop))
    time_cruise_hr_loop = np.zeros(len(V_t_loop))
    W_batt_cruise_loop = np.zeros(len(V_t_loop))
    W_batt_cruise_lb_loop = np.zeros(len(V_t_loop))
    total_mass_loop = np.zeros(len(V_t_loop))
    W_sum_loop = np.zeros(len(V_t_loop))
    W_sum_loop_fuel = np.zeros(len(V_t_loop))

    for element in V_t_loop:
        W_fuel_system_loop[element] = W_fuel_system_fxn(element, V_i, N_t, N_en)
        time_cruise_hr_loop[element] = (V_t_loop[element] - fuel_used_climb) / gal_per_hr
        W_batt_cruise_loop[element] = battery_mass_known_run_time_fxn(time_cruise_hr_loop[element], cruise_power_average , cell_grav_density, effeciency)
        W_batt_cruise_lb_loop[element] = W_batt_cruise_loop[element] * lb_to_kg

        if Pick_Engines == True:
            total_mass_loop[element] = W_sum + W_batt_climb_lb + W_batt_cruise_lb_loop[element]
        else:
            total_mass_loop[element] = W_sum + W_batt_climb_lb
        W_sum_loop[element] = total_mass_loop[element] - W_fuel_system + W_fuel_system_loop[element]
        W_sum_loop_fuel[element] = W_sum_loop[element] + element * 6

    fig3 = plt.figure()
    ax3 = fig3.add_subplot(111)
    ax3.plot(V_t_loop, W_sum_loop_fuel, '-', color='blue') #weight with full fuel
```

```
    #ax3.plot(V_t_loop, W_sum_loop) #weight without fuel
    ax3.plot(V_t_loop, np.full((len(V_t_loop), 1), airplane_mass_lbs), '-', color='red')
    ax3.plot(V_t_loop, np.full((len(V_t_loop), 1), airplane_mass_lbs - 198), '-', color='orange')

    #graph formatting
    ax3.set_xlabel("Fuel Volume, gal")
    ax3.set_ylabel("Weight, lb")
    #ax3.set_title(Plane_List[Plane_Select] + " Fixed fuel capacity")
    ax3.legend(["Aircraft Weight with Fuel", "Max Gross Weight (MGW)", "MGW minus Avg Men Weight"], loc='upper left')
    yaxis_top = max(W_sum_loop_fuel) + max(W_sum_loop_fuel)*0.1
    ax3.set_ylim(0, np.around(yaxis_top/100, decimals=0)*100)
    ax3.set_xlim(0, max(V_t_loop))
    ax3.grid(b=True)

    fig3.savefig(("outputs" + "\\" + Plane_List[Plane_Select] + "_hybrid_fuel_vol" + ".png" ), dpi=400, transparent=False)

##########################################
## CG Analysis
#############################################
CG_analysis = False
if CG_analysis == True:
    arms = np.array([wing_arm, horo_tail_arm, vert_tail_arm, fuselage_arm, main_gear_arm, nose_gear_arm, engine_arm, avionics_arm, wing_tanks, wing_tanks, electrical_arm, front_seat, rear_seat], dtype = "float64")
    W_all = np.array([W_wing, W_horo_tail, W_vert_tail, W_fuselage, W_main_landing_gear, W_nose_landing_gear, W_installed_engine_total, W_avionics, W_fuel_system, W_hydraulics, W_electrical, W_furnishings/2, W_furnishings/2], dtype = "float64")
    moments = arms * np.transpose(W_all)
    moments_names = np.array(['wing', 'horo tail', 'vert tail', 'fuselage', 'main landing gear', 'nose landing gear', 'installed engine total', 'avionics', 'fuel system', 'hydraulics', 'electrical', 'front seat', 'back seat'])
    sum_moments = np.sum(moments)
    cg = sum_moments / W_sum

#File Name : aerodynamic_calcs_fxns.py
import numpy as np

def drag_eqn_fxn(coeff_drag, air_den, velocity, area):
    drag = 0.5 * coeff_drag * air_den * (velocity**2) * area
    return drag

def dynamic_pressure_fxn(air_density, velocity):
        dyn_pressure = 0.5 * air_density * velocity ** 2
        return dyn_pressure

def coeff_lift_fxn(airplane_weight, climb_angle, air_den, wing_ref_area, velocity):
        coeff_lift = (2 * airplane_weight * np.cos(climb_angle))/(air_den * wing_ref_area * velocity**2)
        return coeff_lift

def K_fxn_simple(wing_area, little_e):
        K = 1/(np.pi * wing_area * little_e)
        return K

```

```
def K_fxn(aspect_ratio, CL_alpha, S):
    K100 = 1/(aspect_ratio*np.pi)
    K0 = 1/CL_alpha
    K = S*K100 + (1-S)*K0
    return K

def actuator_disk_fxn(drag_force, airplane_weight, thrust, disk_area, air_den, v):

    if len([thrust]) > 1:
        for element in thrust:
            v_exhaust = np.sqrt((((2*thrust)/(air_den*disk_area))  + (v**2))

    #solve for power expended by actuator disk
            power_out =  ((air_den * disk_area * v)/2) * ((v_exhaust**2) - (v**2))
            power_out_check = thrust * v
            power_out_hp = power_out/ 745.6999

            power_in = thrust * (0.5*(v + v_exhaust))
            power_in_hp = power_in/ 745.6999

            effeciency = power_out/power_in
    else:
        v_exhaust = np.sqrt((((2*thrust)/(air_den*disk_area))  + (v**2))

        #solve for power expended by actuator disk
        power_out =  ((air_den * disk_area * v)/2) * ((v_exhaust**2) - (v**2))
        power_out_check = thrust * v
        power_out_hp = power_out/ 745.6999

        power_in = thrust*(0.5*(v+v_exhaust))
        power_in_hp = power_in/ 745.6999

        effeciency = power_out/power_in

    return [v_exhaust, power_out, power_in, power_out_hp, power_in_hp, power_out_check, effeciency]


def lift_to_drag_cruise_fxn(air_density, velocity, coeff_lift_cruise, Cfe, wet_ref_ratio, aspect_ratio, little_e, wing_load_cruise):
    q = dynamic_pressure_fxn(air_density, velocity)
    CDo = Cfe * wet_ref_ratio

    lift_to_drag = 1/( ((q * CDo)/wing_load_cruise) + (wing_load_cruise * (1/(q * np.pi * aspect_ratio * little_e))) )

    return lift_to_drag

def CL_alpha_fxn(velocity, Cla, aspect_ratio, sweep_angle, medium_SoS, altitude):

    mach_num = velocity/medium_SoS
    beta = np.sqrt(1 - mach_num**2)

    eta = Cla/((2*np.pi)/beta)
```

```
        CLalpha = (2*np.pi*aspect_ratio * (0.98))/(2 + np.sqrt(4 +
(((aspect_ratio**2)*(beta**2))/(eta**2))*(1+((np.tan(sweep_angle))**2)/beta**2)))

        return CLalpha


def airspeed_conversion_fxn(pressure, pressure_sea, air_den_alt, air_den_sea, velocity, CAS_kts, SoS):
  qc = pressure*(((1+0.2*(velocity/SoS)**2)**3.5) - 1)
  EAS_num = (((qc/pressure)+1)**(2/7)) - 1
  EAS_denom = (((qc/pressure_sea)+1)**(2/7)) - 1
  CAS = CAS_kts
  EAS = CAS * np.sqrt(pressure/pressure_sea) * (EAS_num/EAS_denom)**0.5
  TAS = EAS / np.sqrt(air_den_alt/air_den_sea)

  return EAS, TAS
```

```
#File Name: air_properties.py
import numpy as np
from conversion_factors import *

air_den_sea_level = 1.18 #kg/cu.m
air_den_sea_level_slug = 0.00238 #slugs/cu.ft
air_den_sea_level_atm = 1

alt_air = np.array([-1000, 0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 15000,        20000,
        25000, 30000, 40000, 50000, 60000, 70000, 80000], dtype = "float64")
#m

temp_air = np.array([21.5,15,8.5,2,-4.49,-10.98,-17.47,-23.96,-30.45,-36.94,-43.42,-49.9,-56.5,-56.5,-51.6,-46.64,-
22.8,-2.5,-26.13,-53.57,-74.51])
#degC

absolute_press_air = np.array([1.39,10.13,9.772,9.421,9.081, 8.751, 8.431, 8.120, 7.819, 7.527, 7.244, 6.969])
#10^4 N/m^2

density_air = np.array([1.263,1.227,1.191,1.154, 1.124, 1.087, 1.057, 1.027, 0.995, 0.964, 0.933, 0.907])
#kg/m^3

dynamic_vis_air =
np.array([1.821,1.789,1.758,1.726,1.694,1.661,1.628,1.595,1.561,1.527,1.493,1.458,1.422,1.422,1.448,1.475,1.601
,1.704,1.584,1.438,1.321])
#10^-5 Ns/m^2

speed_of_sound_fps = np.array([1120.3, 1116.5, 1112.6, 1108.8, 1104.9, 1101.0, 1097.1, 1093.2, 1089.3, 1085.3,
1081.4, 1077.4])
#ft/s

speed_of_sound = speed_of_sound_fps * fps_to_mps

air_density_ratio = air_den_sea_level/air_den_sea_level
```

```
#File Name: battery_cell_prop.py
#cell stats
cell_volts = 3.6 #volts
cell_mil_amp_hours = 3180 #mAh
cell_amp_hours = cell_mil_amp_hours/1000 #A
cell_grav_density = 217 #Wh/kg
cell_vol_density = 630 #Wh/L
cell_mass_density = (cell_vol_density/cell_grav_density) * 1000 #kg per cu.m
cell_mil_amps = 2980
cell_amps = cell_mil_amps / 1000

#calc mass
#mass = (1/cell_density) * cell_volts * (1/1000) * cell_mil_amp_hours
cell_mass = 49.5 / 1000 #kg

cell_diam_mm = 18.25 #mm
cell_diam = cell_diam_mm/1000 #m
cell_length_mm = 65.10 #mm
cell_length = cell_length_mm/1000

```

```
#File Name: C172airfoil_prop_properties.py
import numpy as np
from physical_constants import *
from conversion_factors import *
from C172dim import *

disk_area = (np.pi * prop_diam**2)/4 #relevant area for

wing_ref_area = wing_area #Sref
wing_ref_area_ft = wing_ref_area * ft_to_m**2

surf_area = plane_area *2
surf_area = np.append(surf_area, plane_area[2]*2)
#horo tail, vert tail, wing, fuselodge -- fuselodge estimated to be equivelant to wing surface area
wetted_area = np.sum(surf_area)
wetted_area_ft = wetted_area * ft_to_m**2

wet_ref_ratio = wetted_area/wing_ref_area

#airfoil properties
Cla_10 = 1.25

def calibration_C172_fxn(IAS):
	CAS = (2e-7)*IAS**4 - 0.0001*IAS**3 + 0.0209*IAS**2 - 0.8836*IAS + 59.416
	return CAS

Vy_IAS = 74
Vy_CAS = calibration_C172_fxn(Vy_IAS)
Vx_IAS = 56
Vx_CAS = calibration_C172_fxn(Vx_IAS)
Va_IAS = 105
Va_CAS = calibration_C172_fxn(Va_IAS)
Vs_IAS = 53
```

```
Vs_CAS = calibration_C172_fxn(Vs_IAS)

Vs_flaps = 48

v_cruise_IAS = 110
v_cruise_CAS = calibration_C172_fxn(v_cruise_IAS)

S = 0.87

#File Name: C712dim.py
import numpy as np
from conversion_factors import *

wing_span = 10.9982 #m
wing_span_ft = wing_span * ft_to_m
overall_length = 8.28 #m
overall_length_ft = overall_length * ft_to_m
height = 2.72 #m
height_ft = height * ft_to_m


sweep_angle_deg = 0 #deg
sweep_angle = sweep_angle_deg * np.pi/180 #rad

wing_area_ft = 174 #sq.ft from C172 POH
wing_area = wing_area_ft / (ft_to_m**2)
horo_tail_in = 6649.465 #in^2
horo_tail = horo_tail_in * in_to_m**2
vert_tail_in = 4034.507 #in^2
vert_tail = vert_tail_in * in_to_m**2
plane_area = np.array([horo_tail, vert_tail, wing_area])
#horo tail, vert tail, wing
plane_area_ft = plane_area * ft_to_m**2

fuselage_area_in = (9908.823 + 9271.58 + 9681.588)
fuselage_area = fuselage_area_in * in_to_m**2
fuselage_area_ft = fuselage_area * ft_to_m**2

aspect_ratio = (wing_span**2) / wing_area

prop_diam_in = 76 #inches
prop_diam = in_to_m * prop_diam_in #convert into meters

MAC_in = 58.80
MAC_m = MAC_in / in_to_m

thickness_in = 6.098

total_fuel = 56.0 #gal

cruise_fuel_consumption = 10 #gal per hour at ~50%
climb_fuel_consumption = 12.8 #gal per hour at 75%

```

```
fuselage_arm = 55 + 70
wing_arm = 55 + 40
horo_tail_arm = 220 + 55
vert_tail_arm = 240 + 55
engine_arm = 55 - 20
#gear_arm_mm = 27
#gear_arm = gear_arm_mm * meas_ratio_b
main_gear_arm = 55 + 55
nose_gear_arm= 55 - 10
avionics_arm = 55 + 20
electrical_arm = 55 + 15
front_seat = 55 + 37
rear_seat = 55 + 73
wing_tanks = 55 + 37

```

```
#File Name: C172weights.py
import numpy as np
from physical_constants import *
from conversion_factors import *

airplane_mass = 1156 #kg
airplane_mass_lbs = airplane_mass * lb_to_kg
airplane_weight = airplane_mass * gravity
airplane_weight_lbs = airplane_weight * lb_to_kg


BEW = 750 #kg
BEW_lbs = BEW * lb_to_kg

engine_weight_lb = 300
engine_weight = engine_weight_lb / lb_to_kg
engine_power_hp = 180
engine_power = engine_power_hp * hp_to_kw


power_loading_lbhp = airplane_weight_lbs / engine_power_hp #lb per hp

```

```
#File Names: conversion_factors.py
import numpy as np

in_to_m = 0.0254 # m per in
ft_to_m = 3.28084 # foot per meter
lb_to_kg = 2.204623 #2.2lbs to 1kg
kts_to_mps = 0.5144447 #meters per second to kts
mps_to_fpm = 196.85 # 1 mps to 196.85 fpm
kts_to_fps = 1.68781 # 1 kts to 1.68781 fps
hp_to_kw = 0.7457 #kw to 1 hp
N_to_lbf = 0.22480894244319 #1N per lbf
slugs_to_lb = 32.174 #32.174 pounds per 1 slug
nm_to_km = 1.852 #1 nautical mile per 1.852 km
lbperhp_to_kgperkw = 0.608277 #1 lb/hp to 0.608277 kg/kw
fps_to_mps = 0.3048 #0.3048mps per 1fps

```

```
#File Name: DA40airfoil_prop_properties.py
import numpy as np
from physical_constants import *
from conversion_factors import *
from DA40dim import *

disk_area = (np.pi * prop_diam**2)/4 #relevant area for

wing_ref_area = wing_area #Sref
wing_ref_area_ft = wing_ref_area * ft_to_m**2


surf_area = plane_area *2
surf_area = np.append(surf_area, plane_area[2]*2)
#horo tail, vert tail, wing, fuselodge -- fuselodge estimated to be equivelant to wing surface area
wetted_area = np.sum(surf_area)
wetted_area_ft = wetted_area * ft_to_m**2

wet_ref_ratio = wetted_area/wing_ref_area


#airfoil properties
Cla_10 = 1.6

Vy_CAS = 67
Vx_CAS = 67
Va_CAS = 108
Vs_CAS = 49

v_cruise_CAS = 110

S = 0.9

```

```
#File Name: DA40dim.py
import numpy as np
from conversion_factors import *

wing_span = 11.94 #m
wing_span_ft = wing_span * ft_to_m
overall_length = 8.01 #m
overall_length_ft = overall_length * ft_to_m
height = 1.97 #m
height_ft = height * ft_to_m

aspect_ratio = 10.53
sweep_angle_deg = 1 #deg
sweep_angle = sweep_angle_deg * np.pi/180 #rad

wing_area = 13.54 #sq.m from diamond POH
wing_area_ft = 145.7 #sq.m from diamond POH
plane_area = np.array([2.34, 1.60, wing_area])
plane_area_ft = np.array([25.2, 17.2, wing_area_ft])
#horo tail, vert tail, wing
```

```
fuselage_area = 13.7
fuselage_area_ft = fuselage_area * ft_to_m**2


prop_diam_in = 70.8 #inches
prop_diam = in_to_m * prop_diam_in #convert into meters

MAC_m = 1.121
MAC_in = 44

total_fuel = 41.2

cruise_fuel_consumption = 7 #gal per hour at ~50%
climb_fuel_consumption = 11 #gal per hour at 75%


meas_ratio_b = 315/122.5 #in per mm

fuselage_arm_mm = 39.478 + 6
fuselage_arm = fuselage_arm_mm * meas_ratio_b
wing_arm = 103.5
horo_tail_arm_mm = 112
horo_tail_arm = horo_tail_arm_mm * meas_ratio_b
vert_tail_arm_mm = 107
vert_tail_arm = vert_tail_arm_mm * meas_ratio_b
#engine_arm_mm = 15
#engine_arm = engine_arm_mm * meas_ratio_b
engine_arm = 39.4
gear_arm_mm = 27
gear_arm = gear_arm_mm * meas_ratio_b
main_gear_arm_mm = 40
main_gear_arm = main_gear_arm_mm * meas_ratio_b
nose_gear_arm_mm = 13
nose_gear_arm = nose_gear_arm_mm * meas_ratio_b
avionics_arm_mm = 29
avionics_arm = avionics_arm_mm * meas_ratio_b
electrical_arm_mm = 26
electrical_arm = electrical_arm_mm * meas_ratio_b
front_seat = 90.6
rear_seat = 128.6
wing_tanks = 103.5

thickness_mm = 2
thickness_in = thickness_mm * meas_ratio_b

```

File Names: DA40weights.py

```
import numpy as np
from physical_constants import *
from conversion_factors import *

airplane_mass = 1150 #kg
airplane_mass_lbs = airplane_mass * lb_to_kg
airplane_weight = airplane_mass * gravity
```

```
airplane_weight_lbs = airplane_weight * lb_to_kg


BEW = 735 #kg
BEW_lbs = BEW * lb_to_kg

engine_weight_lb = 300
engine_weight = engine_weight_lb / lb_to_kg
engine_power_hp = 180
engine_power = engine_power_hp * hp_to_kw

power_loading_lbhp = airplane_weight_lbs / engine_power_hp #lb per hp

#File Name: physical_constants.py
import numpy as np

gravity = 9.807 #m/s

alt_gravity = np.array([-1000, 0, 1000, 2000, 3000, 4000,         5000, 6000,          7000, 8000,          9000, 10000,
15000,    20000,    25000, 30000, 40000, 50000, 60000, 70000, 80000])
#m

gravity_alt =
np.array([9.81,9.807,9.804,9.801,9.797,9.794,9.791,9.788,9.785,9.782,9.779,9.776,9.761,9.745,9.73,9.715,9.684,9.
654,9.624,9.594,9.564])

#File Name: seimens_electric_motor_prop.py
import numpy as np
from conversion_factors import *

#SP70D
motor_volts_SP70D = 400 #v
motor_power_max_SP70D = 92 #kW
motor_power_max_SP70D_hp = motor_power_max_SP70D / hp_to_kw
motor_power_cont_SP70D = 70 #kw
motor_power_cont_SP70D_hp = motor_power_cont_SP70D / hp_to_kw
motor_torque_max_SP70D = 340
motor_torque_cont_SP70D = 260
motor_speed_rpm_SP70D = 2600
motor_peak_eff_SP70D = 0.95
weight_SP70D = 26
weight_SP70D_lbs = weight_SP70D * lb_to_kg


#SP55D
motor_volts_SP55D = 400 #v
motor_power_max_SP55D = 72 #kW
motor_power_max_SP55D_hp = motor_power_max_SP55D / hp_to_kw
motor_power_cont_SP55D = 55 #kw
motor_power_cont_SP55D_hp = motor_power_cont_SP55D / hp_to_kw
motor_torque_max_SP55D = 240
motor_torque_cont_SP55D = 180
motor_speed_rpm_SP55D = 3000
```

```
motor_peak_eff_SP55D = 0.95
weight_SP55D = 26
weight_SP55D_lbs = weight_SP55D * lb_to_kg

#SP260D
motor_volts_SP260D = 580 #v
motor_power_max_SP260D = 260 #kW
motor_power_max_SP260D_hp = motor_power_max_SP260D / hp_to_kw
motor_power_cont_SP260D = 260 #kw
motor_power_cont_SP260D_hp = motor_power_cont_SP260D / hp_to_kw
motor_torque_max_SP260D = 977
motor_torque_cont_SP260D = 1000
motor_speed_rpm_SP260D = 2500
motor_peak_eff_SP260D = 0.95
weight_SP260D = 50
weight_SP260D_lbs = weight_SP260D * lb_to_kg

#SP200D
motor_volts_SP200D = 580 #v
motor_power_max_SP200D = 204 #kW
motor_power_max_SP200D_hp = motor_power_max_SP200D / hp_to_kw
motor_power_cont_SP200D = 204 #kw
motor_power_cont_SP200D_hp = motor_power_cont_SP200D / hp_to_kw
motor_torque_max_SP200D = 1500
motor_torque_cont_SP200D = 1500
motor_speed_rpm_SP200D = 1300
motor_peak_eff_SP200D = 0.95
weight_SP200D = 49
weight_SP200D_lbs = weight_SP200D * lb_to_kg

#File Name: SR22airfoil_prop_properties.py
import numpy as np
from physical_constants import *
from conversion_factors import *
from SR22dim import *

disk_area = (np.pi * prop_diam**2)/4 #relevant area for

wing_ref_area = wing_area #Sref
wing_ref_area_ft = wing_ref_area * ft_to_m**2


surf_area = plane_area * 2
surf_area = np.append(surf_area, plane_area[2]*2)
#horo tail, vert tail, wing, fuselodge -- fuselodge estimated to be equivelant to wing surface area
wetted_area = np.sum(surf_area)
wetted_area_ft = wetted_area * ft_to_m**2

wet_ref_ratio = wetted_area/wing_ref_area

#airfoil properties
Cla_10 = 1.2

```

```
Vy_CAS = 108
Vx_CAS = 88
Va_CAS = 108
Vs_CAS = 74

v_cruise_CAS = 130

S = 0.9

```

```
#File Name: SR22dim.py
import numpy as np
from conversion_factors import *

wing_span = 11.67 #m
wing_span_ft = wing_span * ft_to_m
overall_length = 7.92 #m
overall_length_ft = overall_length * ft_to_m
height = 2.71 #m
height_ft = height * ft_to_m


sweep_angle_deg = 1 #deg
sweep_angle = sweep_angle_deg * np.pi/180 #rad

wing_area = 13.5 #
wing_area_ft = wing_area * ft_to_m**2 #sq.m from diamond POH
aspect_ratio = (wing_span**2)/wing_area

plane_area = np.array([2.34, 1.60, wing_area])
plane_area_ft = plane_area * ft_to_m**2
#horo tail, vert tail, wing
fuselage_area = 26.12
fuselage_area_ft = fuselage_area * ft_to_m**2


prop_diam_in = 78 #inches
prop_diam = in_to_m * prop_diam_in #convert into meters

total_fuel = 94.5

cruise_fuel_consumption = 15 #gal per hour at ~50%
climb_fuel_consumption = 21 #gal per hr

power_loading_lbhp = 11.61 #lb per hp

thickness = 2 * (11.67/12.5)
thickness_in = thickness * 39.3701

MAC_in = 47.7
MAC_m = 1.21

```

```
#File Name: SR22weights.py
import numpy as np
```

```
from physical_constants import *
from conversion_factors import *

airplane_mass = 1633 #kg
airplane_mass_lbs = airplane_mass * lb_to_kg
airplane_weight = airplane_mass * gravity
airplane_weight_lbs = airplane_weight * lb_to_kg


BEW = 952.5 #kg
BEW_lbs = BEW * lb_to_kg

engine_weight_lb = 496
engine_weight = engine_weight_lb / lb_to_kg
engine_power_hp = 310
engine_power = engine_power_hp * hp_to_kw

power_loading_lbhp = airplane_weight_lbs / engine_power_hp #lb per hp

#File Name: WeightsEstimatesFxns.py
import numpy as np
from conversion_factors import *

#eqns
def W_wing_fxn(S_w, W_fw, A, sweep, q, lmbda, thick_to_chord, N_z, W_dg):
	if W_fw == 0:
		W_wing = 0.036 * (S_w**0.758) * ((A/(np.cos(sweep)**2))**0.6) * (q**0.006) * (lmbda**0.04) *
(((100*thick_to_chord)/np.cos(sweep))**(-0.3)) * (N_z * W_dg)**0.49
	else:
		W_wing = 0.036 * (S_w**0.758) * (W_fw**0.0035) * ((A/(np.cos(sweep)**2))**0.6) * (q**0.006)
* (lmbda**0.04) * (((100*thick_to_chord)/np.cos(sweep))**(-0.3)) * (N_z * W_dg)**0.49
	return W_wing

def W_horo_tail_fxn(N_z, W_dg, q, S_ht, thick_to_chord, sweep, A, sweep_ht, lmbda_h):
	W_horo_tail = 0.016*((N_z * W_dg)**0.414) * (q**0.168) * (S_ht**0.896) *
(((100*thick_to_chord)/np.cos(sweep_ht))**(-0.12)) * ((A/(np.cos(sweep_ht)**2))**0.043) * (lmbda_h ** -0.02)
	return W_horo_tail

def W_vert_tail_fxn(H_t_H_v, N_z, W_dg, q, S_vt, thick_to_chord, A, sweep_vt, lmbda_vt):
	#if lambda_vt is less thatn 0.2 use 0.2
	if lmbda_vt < 0.2:
		lmbda_vt = 0.2

	W_vert_tail = 0.073*(1+0.2*(H_t_H_v))*((N_z * W_dg)**0.376) * (q**0.122) * (S_vt**0.873) *
(((100*thick_to_chord)/np.cos(sweep_vt))**(-0.49)) * ((A/(np.cos(sweep_vt)**2))**0.357) * (lmbda_vt ** 0.039)
	return W_vert_tail

def W_fuselage_fxn(S_f, N_z, W_dg, L_t, L, D, q, W_press):
	W_fuselage = 0.052*(S_f**1.086) * ((N_z * W_dg)**0.177) * (L_t ** -0.051) * ((L/D)**(-0.072)) *
(q**0.241) + W_press
	return W_fuselage

def W_landing_gear_fxn(N_l, W_l, L_m):
```

```
    W_main_landing_gear = 0.095*((N_l*W_l)**0.768)*((L_m/12)**0.409)
    W_nose_landing_gear = 0.125*((N_l*W_l)**0.566)*((L_m/12)**0.845)
    #reduce total weight of landing gear by 1.4% if non retractable
    total = W_main_landing_gear + W_nose_landing_gear

    W_landing_gear = total - (total * 0.014)
    return W_landing_gear, W_main_landing_gear, W_nose_landing_gear

def W_installed_engine_total_fxn(W_en, N_en):
    W_installed_engine_total = 2.575*(W_en**0.922)*N_en #includes prop and engine mounts
    return W_installed_engine_total

def W_fuel_system_fxn(V_t, V_i, N_t, N_en):
    if V_t == 0 or N_t == 0:
        W_fuel_system = 0
    else:
        W_fuel_system = 2.49*(V_t**0.726) * ((1/(1+(V_i/V_t)))**0.363)*(N_t**0.242)*(N_en**0.157)

    return W_fuel_system

def W_flight_controls_fxn(L, B_w, N_z, W_dg):
    W_flight_controls = 0.053*(L**1.536) * (B_w**0.371) * ((N_z * W_dg * 10**(-4))**0.80)
    return W_flight_controls

def W_hydraulics_fxn(K_h, W_dg, M):
    W_hydraulics = K_h*(W_dg**0.8) * (M**0.5)
    return W_hydraulics

def W_avionics_fxn(W_uav):
    W_avionics = 2.117*(W_uav**0.933)
    return W_avionics

def W_electrical_fxn(W_fuel_system, W_avionics):
    W_electrical = 12.57*(W_fuel_system + W_avionics)**0.51
    return W_electrical

def W_air_con_and_anti_ice_fxn(W_dg, N_p, W_avionics, M):
    W_air_con_and_anti_ice = 0.265*(W_dg*0.52)*(N_p**0.68)*(W_avionics**0.17)*(M**0.08)
    return W_air_con_and_anti_ice

def W_furnishings_fxn(W_dg):
    W_furnishings = 0.0582*W_dg - 65
    return W_furnishings

#File Name: yasa_electric_motor_prop.py
from conversion_factors import *

#P400 R Series
motor_volts1_400 = 700 #v
motor_power1_max_400 = 160 #kW
motor_power1_max_400_hp = motor_power1_max_400 / hp_to_kw
motor_power_cont_400 = 100 #kw
motor_power_cont_400_hp = motor_power_cont_400 / hp_to_kw
```

```
motor_peak_eff_400 = 0.96
weight_400 = 24
weight_400_lbs = weight_400 * lb_to_kg

#750R series
motor_volts1_750 = 350
motor_volts2_750 = 700
motor_power1_max_750 = 100 #kW
motor_power1_max_750_hp = motor_power1_max_750 / hp_to_kw
motor_power2_max_750 = 200 #kW
motor_power2_max_750_hp = motor_power2_max_750 / hp_to_kw
motor_power_cont_750 = 70 #kw
motor_power_cont_750_hp = motor_power_cont_750 / hp_to_kw
motor_peak_eff_750 = 0.96
weight_750 = 37
weight_750_lbs = weight_750 * lb_to_kg
```