Submitted, 2012 Conference on Decision and Control (CDC)
http://www.cds.caltech.edu/~murray/papers/xtm12-cdc.html

Reactive Protocols for Aircraft Electric Power Distribution

Huan Xu, Ufuk Topcu, and Richard M. Murray

***Abstract*— The increasing complexity of electric power systems leads to integration and verification challenges. We consider the problem of designing a control protocol for the aircraft electric power system that meets these system requirements and reacts dynamically to changes in internal system states. We formalize these requirements by translating them into a temporal logic specification language describing the correct behaviors of the system, and apply formal methods to automatically synthesize a controller protocol that satisfies these overall properties and requirements. Through an example, we perform a design exploration to show the benefits and tradeoffs between centralized and distributed control architectures.**

I. INTRODUCTION

Advances in electronics technology has made the transition from conventional to more-electric aircraft (MEA) architectures possible. The concept of electric aircraft is not new; although it was considered by military aircraft designers during World War II, the idea was never implemented due to lack of electric power generation capabilities at that time [1]. Conventional architectures utilize a combination of mechanical, hydraulic, electric, and pneumatic subsystems. The move towards MEAs increases efficiency by reducing power take-offs from the engines that would otherwise be needed to run hydraulic and pneumatic components. Moreover, use of electric systems provides opportunities for system-level performance optimization and decreases life-cycle costs.

Efforts have been made to re-use previously developed systems from conventional aircraft [2], but additional high-voltage networks and electrically-powered components increase the system's complexity, and new approaches need to be considered. These electric power system designs must behave according to certain properties or requirements determined by physical constraints or performance criteria. Because safety of the aircraft is solely or mostly dependent on electric power, the electric power system needs to be highly reliable, fault tolerant, and autonomously controlled. Past work has focused on the analysis of aircraft performance and power optimization by using modeling libraries and simulations [3], [4], [5]. Analysis of all faults or errant behaviors in these models is difficult due to the high complexity of these systems. This has led to a greater emphasis on the use of formal methods to aid in safety and performance certification.

Controllers for an electric power system must be designed so that the system satisfies certain safety and reliability properties. These requirements, however, are typically text-based lists, oftentimes ambiguous in intent or inconsistent with each other. The process of verifying the correctness of a system with respect to these specifications is expensive, both in terms of cost and time. In this paper, we "specify and synthesize" a solution to the design problem In this approach, we begin by converting test-based system specifications for an electric power system into a mathematical formalism using a temporal logic specification language. From these specifications, we then automatically synthesize centralized and distributed controllers, and examine design tradeoffs between these different control architectures.

The remainder of the paper is structured as follows: We describe a standard electric power system, including components, connectivity, and typical design considerations in Section II. Section III details the problem description, including types of specifications and the overall synthesis problem, and is followed by Section IV, which gives a technical description of specification language and synthesis procedure. Sections V and VI present a case study of an electric power system, including variables and formal specifications and presents results for a centralized and distributed control architecture, and is followed by concluding remarks.

II. ELECTRIC POWER DISTRIBUTION SYSTEM

The standard electric power system for a passenger aircraft comprises a certain number of generators (e.g., one or two on the left and right sides of the aircraft) that serve as primary power sources. These generators supply power to a set of loads through dedicated AC buses. Typically, each AC bus delivers power to a DC bus through a transformer rectifier unit. Contactors are high-power switches that can control the flow of power by reconfiguring the topology of the electric power system and can establish connections between components. In the case of a generator failure, an auxiliary power unit (APU) or battery may be used to power buses through a different reconfiguration of system components. These different reconfigurations of the system will change the open or closed status of contactors and thereby affect the power level of different buses or loads.

Next-generation aircraft will have increased safety-criticality reliant on the electric power system and increased number of overall components in the electric power system, raising the complexity of design. The number of configurations quickly goes beyond currently available verification and testing capabilities. In this paper, we investigate an alternative way for the design of control protocols for electric power systems on more-electric aircraft, and use the sample electric power system in Fig. 1 as a running example.

The authors are with the California Institute of Technology. mumu@caltech.edu

A. System Components

The electric power system schematic in Fig. 1 includes a combination of generators, contactors, buses, and loads. The following is a brief description of the components referenced in the primary power distribution single-line diagram [6].

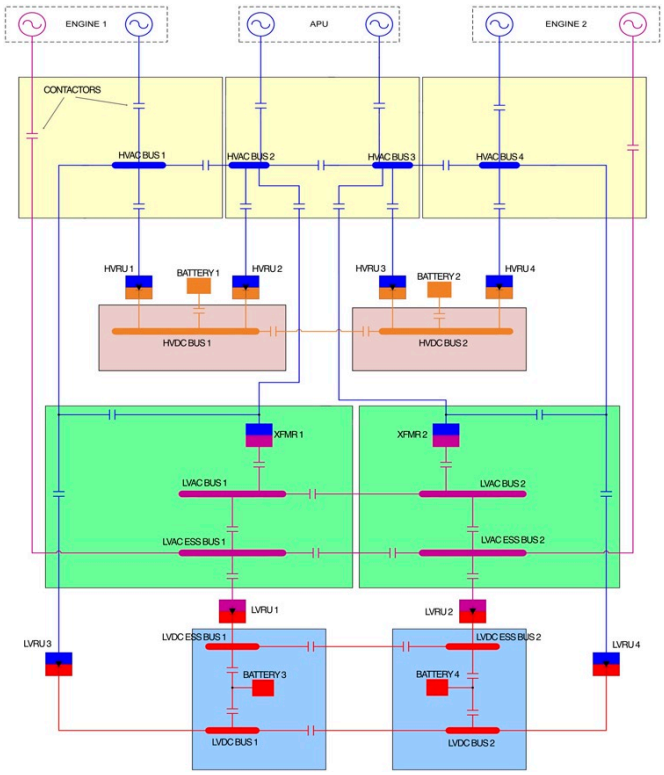

Fig. 1. Single line diagram of an electric power system adapted from a Honeywell, Inc. patent [7]. Two high-voltage generators, APUs, and low-voltage generators serve as power sources for the aircraft. Depending on the configuration of contactors, power can be routed from sources to buses through the contactors, rectifier units, and transformers. Buses are connected to subsystem loads. Batteries can be used to provide emergency backup power to DC buses. A high-resolution figure can be viewed at http://www.cds.caltech.edu/ utopcu/misc/SLD.pdf.

Buses: AC and DC power buses for both high and low-voltage deliver power to a number of buses, loads, or power conversion equipment. Buses can be essential or non-essential. Essential buses supply loads which should always remain powered, while non-essential buses supply loads which may be shed in the case of a fault.

Generators: AC generators supply power to buses, and can operate at either high or low-voltages.

Contactors: Contactors are electronic switches that connect the flow of power from sources to buses and loads (represented by $\dashv\vdash$). They can reconfigure (i.e., switch between open and closed) through one or multiple controllers.

Transformer Rectifier Units: Rectifier units convert AC power to DC power (e.g., HVRU in Fig. 1.) Transformers step down a high-voltage to a lower one (e.g., XFMR in Fig. 1.) A combination Transformer Rectifier unit both decreases voltage and converts it from AC to DC power (e.g., LVRU.)

Batteries: Electrical storage medium used to provide short-term power during emergency conditions.

B. System Description

The following provides a brief description of the electric power system topology in Fig. 1.

- At the top are six AC generators: two low-voltage, two high-voltage, and two APUs.
- The three panels below the generators contain the high-voltage AC distribution system. Each panel represents the physical separation of components within the aircraft. We denote components that can connect or disconnect from each other through the opening or closing of contactors as selectively connected.
- Selectively connected to the four high-voltage AC buses are four rectifier units (HVRU) which transform AC to DC power. Each high-voltage DC bus also has a battery source which can also be selectively connected.
- High-voltage AC Buses 2 and 3 are selectively connected to a set of transformers (labeled as XFMR) that convert high-voltage AC power to low-voltage AC power. The low-voltage AC system is depicted in the two panels in Fig. 1 right below the high-voltage AC panels. The transformers are connected to a set of four low-voltage AC buses. LVAC ESS Bus 1 and LVAC ESS Bus 2 are essential, and are selectively connected to the two low-voltage AC emergency generators.
- Low-voltage AC essential buses are directly connected to low-voltage rectifier units (LVRU). There are four low-voltage DC buses and batteries that may also be selectively connected. Power can also be routed from high-voltage AC buses through transformers to LVDC Main Bus 1 and 2.

The control protocol design problem considers how the system shall reconfigure as a function of the changes in flight conditions and faults in the components. We focus on the problem of dynamic reconfiguration of the primary distribution system, which involves the start-up or shut-down of high-voltage generators and APUs as well as the reconfiguration of contactors in order to route power to high-voltage buses and loads.

III. Problem Description

Given a topology of an electric power system, the main design problem is determining all correct configurations of contactors for all flight conditions and faults that can occur. For a configuration to be "correct" means that it satisfies system requirements, also referred to as specifications. We now discuss a few sample specifications relevant to the problems found in Fig. 1.

A. Specifications

Specifications are generally expressed in terms of safety, performance, and reliability properties.

Safety: These specifications constrain the way each bus can be powered and the length of time it can tolerate power shortages. In order for AC generators to work in parallel with each other, they need to match their respective voltages and frequencies. A mismatch can lead to generator damage. To avoid difficulties in synchronization, we disallow any

paralleling of AC sources (i.e., no bus should be powered by multiple AC generators at the same time.) Essential loads, such as flight-critical actuators, are connected to essential AC and DC buses. These loads should never be unpowered for more than 50 msec. Lastly, the time it takes for contactors to switch configurations will vary due to physical hardware constraints. Typical opening times can range between 10-20 msec, while closure times are between 15-25 msec [6].

Performance: Performance specifications rank desired system configurations. A generator priority list is assigned to each bus specifying the order of sources each bus should be powered . If the first priority generator is unavailable, then it will be powered from the second priority generator, etc. A hypothetical prioritization list is shown in Table I for HVAC Bus 1. For bus 1, G_L is the first priority on the list. if the left high-voltage generator is healthy, then bus 1 receives power from that generator. If G_L is faulty, then Bus 1 should receive power its second priority G_R, and so forth.

TABLE I
SOURCE PRIORITY TABLE FOR HVAC BUSES

Priority	Bus 1	Bus 2	Bus 3	Bus 4
1	G_L	A_L	A_R	G_R
2	G_R	G_L	G_R	G_L
3	A_L	G_R	G_L	A_R
4	A_R	A_R	A_L	A_L

Reliability: These specifications describe the bounds on probability of failures within the system. Every component comes with a reliability level. A level ϵ of reliability, for example, indicates that one failure will occur every $\frac{1}{\epsilon}$ hours. Given multiple component failures, systems should be designed to tolerate any combination of component faults that has a joint probability of less than a certain pre-specified level. Practically, these reliability specifications determine the combination of simultaneous faults that need to be accounted for by the control protocol. An electric power system should still be able to satisfy its safety specifications given any combination of faults that lead to the pre-specified level. (In the design procedure proposed in subsequent sections, implicitly account for reliability specifications through the environment assumptions.)

B. The Synthesis Problem

The overall goal of the design problem is synthesizing a control protocol that, when implemented on the electric power system, ensures that the controlled system satisfies the specifications discussed previously. Roughly speaking, contactors are the actuators that can be controlled by the system. In other words, the system reconfigures the distribution topology and the paths through which the bus is powered by opening and closing the contactors. The correctness of the system, on the other hand, is not merely a function of the states of the controlled variables. It needs to be interpreted in conjunction with the statuses of the externalities that interact with the system yet cannot be controlled.

On the other hand, it is necessary to incorporate the information on the potential environment conditions under which the system is expected to operate. If the environment variables are not properly constrained, then the resulting control protocol may be overly conservative, and it and may not be possible to construct a protocol that ensures the satisfaction of the system requirements. In other words, an essential component of the protocol synthesis problem is the assumptions which specify what environment behaviors the controller shall correctly react to. Consequently, the overall goal is to design a protocol that determines how controlled variables shall move at each point of the execution as a function of the behaviors of all system variables as long as the environment assumptions are satisfied.

One of the main limitations in common practice is that specifications are written in languages (e.g., English) that are not mathematically suitable for computational analysis or design. The verification of correctness is left for post-design simulations and tests. The resulting control protocols are oftentimes quite complicated for formal reasoning and not suitable because of the lack of formal specifications. We pursue a complementing approach, namely "formally specify and then design." Potential benefits of this change in the strategy for establishing the correctness of the controllers include alleviating any ambiguity (and potentially even inconsistency) in the specifications and partially automating the design procedure. In the next section, we discuss a candidate formal specification language and means for synthesizing reactive control protocols from specifications expressed in this language.

IV. FORMAL SPECIFICATION AND SYNTHESIS

We now discuss a formal specification language utilized for the synthesis of control protocols later in this section.

A. Formal Specification Using Linear Temporal Logic

In reactive systems (i.e., systems which react to an environment), correctness will depend on, not only inputs and outputs of a computation, but on execution of the system. Temporal logic is a branch of logic that incorporates temporal aspects to reason about propositions in time, and was first used as a specification language by Pneuli [8]. In this paper, we consider a version of temporal logic called linear temporal logic (LTL) [9]. Before describing LTL, we first define an atomic proposition, LTL's main building block.

Definition 1: A system consists of a set V of variables. The domain of V, denoted by $dom(V)$, is the set of valuations of V.

Definition 2: An *atomic proposition* is a statement on system variables v that has a unique truth value (*True* or *False*) for a given value v. Let $v \in dom(V)$ be a state of the system and p be an atomic proposition. Then $v \models p$ if p is *True* at the state v. Otherwise, $v \not\models p$.

LTL also includes Boolean connectors like negation ($\neg$), disjunction ($\vee$), conjunction ($\wedge$), material implication ($\rightarrow$), and two basic temporal modalities *next* ($\bigcirc$) and *until* ($\mathcal{U}$). By combining these operators, it is possible to specify a wide range of requirements. For a set π of atomic propositions, any atomic proposition $p \in \pi$ is an LTL formula. Given LTL

formulas φ and ψ over π, $\neg\varphi$, $\varphi \vee \psi$, $\bigcirc\varphi$ and $\varphi \,\mathcal{U}\, \psi$ are also LTL formulas. Given a set of variables (i.e., a system) and a set π of atomic propositions in terms of the valuations of these variables (i.e., states of the system). LTL formulas over π are interpreted over infinite sequences of states. Formulas involving other operators can be derived from these basic ones, including *eventually* ($\Diamond$) and *always* ($\Box$).

Let $\sigma = v_0 v_1 v_2 \ldots$ be an infinite sequence of valuations of variables in V and φ be an LTL formula. We say that φ holds at position $i \geq 0$ of σ, written $v_i \models \varphi$, if and only if φ holds for the remainder of the sequence starting at position i. Then, a sequence σ satisfies φ, denoted by $\sigma \models \varphi$, if $v_0 \models \varphi$. Let Σ be the collection of all sequences σ such that $\sigma \in \Sigma$. Then, a system composed of the variables V is said to satisfy φ, written $\Sigma \models \varphi$, if all sequences satisfy φ. (See [9] for more details.)

B. Reactive Synthesis

Let E and P be sets of environment and controlled variables, respectively. Let $s = (e, p) \in dom(E) \times dom(P)$ be a state of the system. Consider a LTL specification φ of assume-guarantee form

$$\varphi = (\varphi_e \rightarrow \varphi_s), \tag{1}$$

where, roughly speaking, φ_e characterizes the assumptions on the environment and φ_s characterizes the system requirements. The synthesis problem is then concerned with constructing a strategy (i.e., a partial function $f : (s_0 s_1 \ldots s_{t-1}, e_t) \mapsto p_t$) which chooses the move of the controlled variables based on the state sequence so far and the behavior of the environment so that the system satisfies φ_s as long as the environment satisfies φ_e. The synthesis problem can be viewed as a two-player game between and environment that attempts to falsify the specification in (1) and a controlled plant that tries to satisfy it.

For general LTL, the synthesis problem has a doubly exponential complexity [10]. A subset of LTL, namely generalized reactivity (1) (GR(1)), can be solved in polynomial time (polynomial in the number of valuations of the variables in E and P) [19]. GR(1) specifications restrict φ_e and φ_s to take the following form, for $\alpha \in \{e, s\}$,

$$\varphi_\alpha := \varphi^\alpha_{\text{init}} \wedge \bigwedge_{i \in I^\alpha_1} \Box\varphi^\alpha_{1,i} \wedge \bigwedge_{i \in I^\alpha_2} \Box\Diamond\varphi^\alpha_{2,i},$$

where $\varphi^\alpha_{\text{init}}$ is a propositional formula characterizing the initial conditions; $\varphi^\alpha_{1,i}$ are transition relations characterizing safe, allowable moves and propositional formulas characterizing invariants; and $\varphi^\alpha_{2,i}$ are propositional formulas characterizing states that should be attained infinitely often.

Given a GR(1) specification, the digital design synthesis tool implemented in JTLV (a framework for developing temporal verification algorithm) [11] generates a finite-state automaton that represents a switching strategy for the system. The Temporal Logic Planning (TuLiP) Toolbox, a collection of Python-based code for automatic synthesis of correct-by-construction embedded control software as discussed in provides an interface to JTLV [12]. For examples discussed in this paper, we use TuLiP.

C. Distributed Synthesis

As discussed earlier, control architectures for electric power systems on more-electric aircraft will likely have distributed structures. We follow the exposition in [13]. For ease of representation, consider the case where the system is composed of two subsystems and the set of variables and global specification $\varphi_e \rightarrow \varphi_s$ decomposed as follows:

Let $\varphi_e, \varphi_{e_1}, \varphi_{e_2}, \varphi_s, \varphi_{s_1}$, and φ_{s_2} be LTL formulas containing variables only from their respective sets of environment variables E, E_1, E_2 and system variables S, S_1, S_2. If the following conditions hold: (1) any execution of the environment that satisfies φ_e also satisfies $(\varphi_{e_1} \wedge \varphi_{e_2})$, (2) any execution of the system that satisfies $(\varphi_{s_1} \wedge \varphi_{s_2})$ also satisfies φ_s, and (3) there exist two control protocols that make the local specifications $(\varphi_{e_1} \rightarrow \varphi_{s_1})$ and $(\varphi_{e_2} \rightarrow \varphi_{s_2})$ true. Then, by a result in [13], implementing these two control protocols together leads to a system where the global specification $\varphi_e \rightarrow \varphi_s$ is met.

Two factors should be taken into account when choosing local environment and system variables e_1, e_2, s_1, and s_2. The first is the number of variables involved in the local synthesis problems. If the possible valuations of variables involved in local specifications are substantially less than the possible valuations of the variables in the global specication, then distributed synthesis would be computationally more efcient than the centralized one (assuming the lengths of LTL formulas for the global and the local specications are of the same order). The second is the conservatism of the distributed synthesis. It is possible that even if the centralized problem is realizable, the local distributed synthesis may be unrealizable. Indeed, let sets of executions be defined as:

$$\Sigma_e = \{\sigma | \sigma \models \varphi_e\}; \quad \Sigma_{e'} = \{\sigma | \sigma \models (\varphi_{e1} \wedge \varphi_{e2})\};$$
$$\Sigma_s = \{\sigma | \sigma \models \varphi_s\}; \quad \Sigma_{s'} = \{\sigma | \sigma \models (\varphi_{s1} \wedge \varphi_{s2})\}.$$

Condition 1 implies that $\Sigma_{e'} \supseteq \Sigma_e$, whereas condition 2 implies that $\Sigma_{s'} \subseteq \Sigma_s$. Local variables and specifications should be chosen so that conditions 1 and 2 are satisfied. Moreover, the conservatism can be reduced by choosing φ_{e_j} and φ_{s_j} such that $\Sigma_{e'}$ is as "small" as possible, and the set $\Sigma_{s'}$ is as "large" as possible in the sense of set inclusion. See Section VI-B.2 for an example of such a refinement and [13] for more details.

V. Synthesis of Reactive Protocols for Aircraft Electric Power Distribution

We address the problem of primary distribution in an electric power system by examining a simplified version of the single-line diagram. Fig. 2 shows the portion of the single-line diagram considered for the problem formulation used in the rest of this paper. This topology consists of the basic high-voltage AC components: two generators and two APUs connect to four buses via seven contactors.

A. Variables

Variables used in this formulation can be classified as environment, controlled, or dependent.

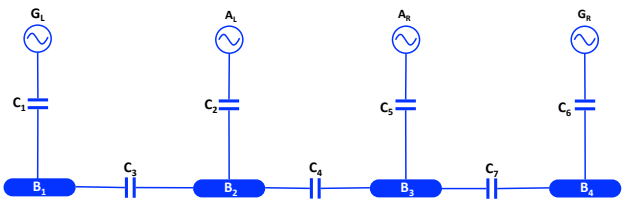

Fig. 2. Simplified single-line diagram used in the centralized problem. Four power sources connect to four buses through a series of seven contactors

- Environment: The health statuses of the left and right generators (G_L, G_R) and APUs (A_L, A_R) can each take values of healthy (1) and unhealthy (0). These statuses are uncontrollable and may change at any point in time.
- Controlled: Contactors connecting generators and APUs to buses, (C_1, C_2, C_5, C_6), can each take values of open (0) or closed (1). A closed contactor will allow power to pass through, while an open one does not. Contactors between buses (C_3, C_4, C_7) can take three values. A value of 0 again denotes an open contactor. A value of -1 or 1 signifies a contactor is closed and that power is flowing from right to left, or left to right, respectively.
- Dependent: Buses B_1, B_2, B_3, and B_4 can be either powered (1) or unpowered (0). Bus values will depend on the status of their neighboring contactors, buses, as well as the health status of connecting generators or APUs.

Timing considerations play a key part in the specifications for an electric power system (as discussed in the next section). LTL, however, only addresses the notion of temporal ordering of events. It can not explicitly incorporate requirements on time intervals. In order to reconcile this discrepancy, the variable $\tilde{C}_i$ for $i \in [1, 7]$ is introduced to represent the controller intent for contactor C_i. The intent variable $\tilde{C}_i$ can take the values as contactor status C_i of open or closed. If a fault occurs, the controller sets the intent for a contactor based on the status of its neighboring generator or bus. An action on contactor status occurs non-deterministically either immediately or one time step later.

B. Formal Specifications

Given the topology in Fig. 2, the following lists the temporal logic specifications used.

Environment Assumption: At least one power source is healthy at any given time.

- $\square\{(G_L = 1) \vee (A_L = 1) \vee (A_R = 1) \vee (G_R = 1)\}$

Power Status of Buses: A bus can only be powered if a contactor is closed and its connecting generator, APU, or neighboring bus is powered. If B_1 is powered if one of two properties holds: G_L is healthy and C_1 is closed, or B_2 is powered and C_3 is closed. If neither of these two are true, then bus B_1 will be unpowered. Specifically:

- $\square\{((C_1 = 1) \wedge (G_L = 1)) \rightarrow (B_1 = 1)\}$
- $\square\{((B_2 = 1) \wedge (C_3 = -1)) \rightarrow (B_1 = 1)\}$
- $\square\{\neg((C_1 = 1) \wedge (G_L = 1)) \vee ((B_2 = 1) \wedge (C_3 = -1)) \rightarrow (B_1 = 0).\}$

A similar set of specifications is applied for B_2, B_3, and B_4.

No Paralleling of AC Sources: One way to avoid paralleling is to explicitly enumerate and eliminate all bad configurations in which buses can be powered from multiple sources. In Fig. 2, for example, paralleling can occur if G_L and A_L are both healthy, and contactors C_1, C_2, and C_3 are all closed. A simple specification would then be to disallow C_3 to be open if both C_1 and C_2 were closed. This "global" approach becomes difficult to scale, however, when the number of paths and components grows large. Specifications need to be written for each combination of generators and contactor paths.

We take a "localized" view on non-paralleling specifications. Instead of examining entire paths, we focus on the source of power coming into each bus. To this end, we first introduce "power flow direction" to contactors C_3, C_4, and C_7. The contactors connecting generators and APUs are strictly unidirectional. We restrict the value of contactors based on the direction in which power may flow depending on the health and status of surrounding buses/sources. For these bidirectional contactors, if the neighboring two nodes (either a generator, APU or bus) is unpowered, then the contactor cannot direct power in the opposite direction those nodes. Note that directionality within the contactor is not present in the physical implementation of hardware. A contactor is either open or closed. The notion of directionality is internal to the problem formulation in order to take a "localized" approach; the physical hardware can only be set to open or closed and has no method of determining the direction of power flow.

If the left generator is unhealthy, then contactor C_3 cannot direct power from left to right, and the intent variable $\tilde{C}_3$ should be assigned accordingly. If the following properties are not true: (1) the left generator is healthy an bus B_2 is powered, or (2) Buses B_2 and B_3 are powered, then contactor C_3 cannot direct power from right to left. This can be written:

- $\square\{\neg(G_L = 1) \rightarrow \neg(\tilde{C}_3 = 1)\}$
- $\square\{\neg(((G_L = 1) \wedge (B_2 = 1)) \vee((B_3 = 1) \wedge (B_2 = 1))) \rightarrow \neg(\tilde{C}_3 = -1).\}$

Given direction of flow in contactors, we can examine examine each bus and eliminate any configuration of contactors which may allow for paralleling of sources. For example, the following configurations are not allowed for bus B_2.

- $\square\{\neg((C_2 = 1) \wedge (C_3 = 1))\}$
- $\square\{\neg((C_2 = 1) \wedge (C_4 = -1))\}$
- $\square\{\neg((C_3 = 1) \wedge (C_4 = -1))\}$.t

Safety-Criticality of Buses: Certain buses connected to safety-critical loads (e.g., flight actuators or de-icers) need to remain powered. Buses also need to be able to be unpowered for short lengths of time in order to reconfigure power sources without violating the non-paralleling specification. In this problem we consider buses B_1 and B_4 to be safety-critical buses, and can be unpowered for no longer than five time steps. This notion of time is implemented through an additional clock variable t for each bus, where each "tick" of the clock represents 10 msec. A safety specification for B_1 would be: If bus B_1 is unpowered, then at the next time

step clock t_1 increases $\square\{(B_1 = 0) \rightarrow (\bigcirc t_1 = t_1 + 1)\}$. If bus B_1 is powered, then at the next time step reset clock t_1 $\square\{(B_1 = 1) \rightarrow (\bigcirc t_1 = 0)\}$. Then, ensure that bus B_1 is never unpowered for more than 5 steps $\square\{t_1 \leq 5\}$.

Unhealthy Buses: A bus connected to an unhealthy source will create a short-circuit failure, leading to excessive electrical currents, overheating, and possible fires. We require that a contactor open when a generator or APU becomes unhealthy to avoid such failures. An example specification for the intent of contactor C_1 would be: $\square\{(G_L = 0) \rightarrow (\tilde{C}_1 = 0)\}$. **Prioritization:** We thus introduce the notion of prioritization on power sources. Generators G_L and G_R, if healthy, will always be connected and used to power left and right side buses, respectively. APUs A_L and A_R are only connected if their respective left and right generator is unhealthy. This corresponds to a notion of nearest generator (in distance). In the below example, contactor C_2 is only closed if the left generator goes unhealthy. This can be written as: $\square\{((G_L = 0) \ \wedge \ (A_L = 1)) \rightarrow (\tilde{C}_2 = 1)\}$.

VI. Results

This section presents some preliminary results for the formal reactive synthesis of control protocols in an electric power system for centralized and distributed controllers.

A. Centralized Controller Design

Fig. 3 shows the simplified single-line diagram overlaid with a sample simulation run. The horizontal axis of each graph in in the figure represents the step of the simulation, starting at step 0 and ending with step 5.

The four graphs in row 1 correspond to the statuses of the environment variables. These values are arbitrarily input, subject to the environment assumption. At each step, generators and APUs can switch between healthy and unhealthy as long as at least one source remains healthy. Graphs in rows 2 and 3 correspond to the contactor statuses generated from the synthesized control protocol. Because power can only flow from a generator or APU, the graphs for the contactors shown in row 2 can only take values of open or closed. Row 3 graphs, however, can take three values corresponding to open, closed with power directed to the right, or closed with power flowing to the left. Graphs in row 4 correspond to the four buses, and the vertical axis represents the power status of the bus. Because buses are dependent variables, these values are determined by the environment variables as well as the contactor configurations.

To better understand the results shown in Fig. 3 let us examine the simulation graphs for a single step, namely step 2. The left generator G_L is unhealthy and contactor C_1 is open. The left APU A_L is healthy, and contactor C_2 is closed. Bus B_2 is powered because it is connected to A_L. Bus B_1 is unpowered because both neighboring contactors C_1 and C_3 are open. Meanwhile, the right generator G_R is healthy and contactor C_6 is closed. Therefore, according to the second set of specifications from Section V-B, bus B_4 is powered. Note, however, that C_5 remains closed even though the right APU is unhealthy. In the previous step, A_R was healthy, and its intent to open $\tilde{C}_5$ in step 2 does not get implemented until step 4. In order to ensure non-paralleling of sources, contactor C_7 must remain open at step 2 because C_5 is closed, even though no power is flowing from the APU. As a result, bus B_3 is unpowered.

Safety-critical buses B_1 and B_4 are never unpowered for more than two time steps throughout the entire simulation sequence. This specification is not imposed on the middle two buses, however, and and thus B_3 can remain unpowered for five steps without violating any system requirements. In addition, at no time in the simulation run are AC sources paralleled. Consider, for example, power flowing to bus B_1. When contactor C_1 is closed (steps 0,1, and 4), C_3 is always open.

B. Distributed Control Architecture

In the following section, we decompose the centralized electric power system topology into two smaller subsystems and synthesize two local controllers. When implemented together, these controllers are guaranteed to be correct with respect to the global specification. The physical decomposition of the electric power system is shown in Fig. 4. Let S_r represent the right subsystem (enclosed in the dotted lines) and S_l represent the left subsystem. The environment and system variables for S_l and S_r are denoted by e_r, s_r, e_l and s_l, respectively. Based on the refinement technique mentioned in Section IV-C, the global specification discussed in Section V-B is satisfied if the following are true:

$$\phi_r \wedge \varphi_{e_l} \rightarrow \varphi_{s_l} \wedge \phi_l, \phi_l \wedge \varphi_{e_r} \rightarrow \varphi_{s_r} \wedge \phi_r, \tag{2}$$

where formulas ϕ_r and ϕ_l represent additional assumptions and guarantees made at the interface between the left and the right subsystems in order to ensure that the global system is realizable. ϕ_l is a guarantee from subsystem S_l and seen as an environment assumption by the controller for subsystem S_r. Similarly, ϕ_r is a guarantee from S_r and an environment assumption in S_l. Specifications for these interface refinements will be stated in the following.

We now present results for two different types of distributed control architectures: master/slave and bidirectional.

1) Master/Slave Control Architecture: For a master/slave architecture, power flow between the decomposed systems is controlled by one side, and unidirectional only. For the decomposition shown in Fig. 4, the subsystem S_r is the "master" and can control the supply of power that can flow from right to left via contactor C_4. Subsystem S_l is the "slave" and can only receive power when S_r provides it.

We decompose the global environment assumption, in which at least one power source must remain healthy at each step, such that $\varphi_{e_r} = \square(A_R = 1 \vee G_R = 1)$ and $\varphi_{e_l} = true$. This ensures that for any execution $\sigma \in \Sigma$, the controller for S_r is able to supply power to S_l at any step. The left generator and APU health statuses are sent to the right side via a health variable H_1. The variable is set to 0 if neither source is healthy, and is set of 1 if either G_L or A_L is healthy such that φ_{e_r} can assume knowledge about the health status of the left side.

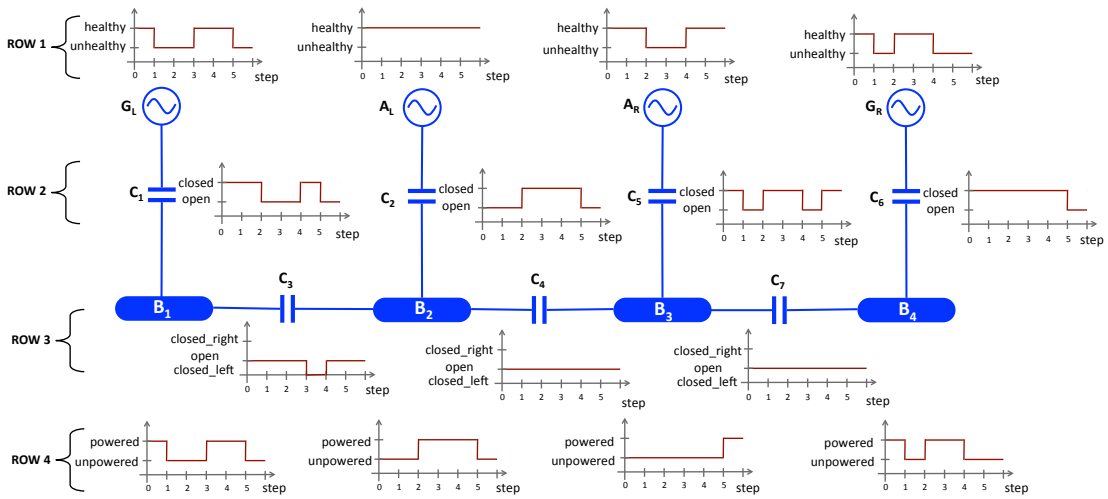

Fig. 3. A simulation result for a centralized controller for the electric power system. The horizontal axis represents the simulation step. Row 1 shows the environment inputs for generator and APU health. Based on these values, the controller values for contactors are set to either open or closed, as seen in Row 2. Additionally, Row 3 shows the direction of power flow through contactors C_3, C_4, and C_7. Row 4 shows the power status for all buses.

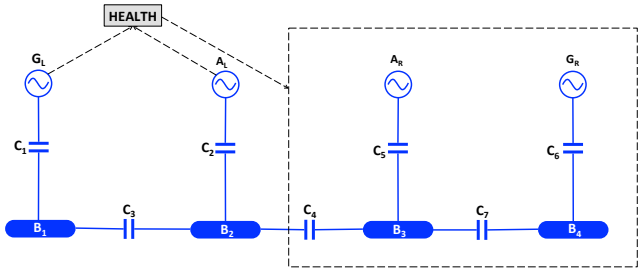

Fig. 4. A distributed controller decomposition for the electric power system. The left hand side sees contactor C_4 as an environment variable, provides the health status of its generator and APU as information to the right side. The right side has control of C_4, which enables the flow of power between the two subsystems.

In order for the master/slave distributed synthesis problem to become realizable, additional assumptions and guarantees (i.e., interface refinements) need to be implemented. It is not enough for power sources G_R and A_R to be able to generate power at all steps. The controller for S_r must also be able to guarantee that power can be delivered to the left. Thus, we introduce ϕ_r as a guarantee for the S_r controller, and as an assumption for S_l controller. Because the master controls the flow of power, a single-sided refinement is sufficient for the design problem to be realizable, and we can set $\phi_l = true$. The additional specification ϕ_r imposes conditions on contactor C_4 and bus B_3 (the components nearest to the interface of S_r and S_l. These specifications are

- Bus B_3 is never unpowered for a set number of time steps n. Essentially, B_3 becomes a safety-critical bus, and we introduce a clock variable t_3 to monitor the power status. $\square\{(B_3 = 0) \rightarrow (\bigcirc t_3 = t_3 + 1)\} \wedge \square\{(B_3 = 1) \rightarrow (\bigcirc t_3 = 0)\} \wedge \square\{t_3 \leq n\}$
- If health status $H_1 = 0$ (i.e., both G_L and A_L are unhealthy), then whenever B_3 is powered, C_4 will close. $\square\{((H_1 = 0) \wedge (B_3 = 1)) \rightarrow (\tilde{C}_4 = -1)\}$

A similar modification must be made for the case where unidirectional power flows from S_l to S_r. In both of the cases discussed in the master/slave architecture, all other specifications remain the same as those discussed from Section V-B and decomposed with their respective components. Simulation results are comparable to those for the centralized controller, shown in Fig. 3, and thus omitted.

2) Bidirectional Power Flow Control Architecture: Consider again the physical decomposition shown in Fig. 4, where power is allowed to flow from either left to right, or right to left. The physical actuation of middle contactor C_4 is still controlled by the right side. The environment variables for S_l include G_L, A_L, and C_4, while environment variables for S_r contain G_R, A_R, B_2, and H_1. Note that this differs from the master/slave control architecture with the necessary addition of B_2 as an environment variable to allow for power to flow in two directions.

The case where there is power flow between S_l and S_r corresponds to a feedback interconnection where part of the output of each system acts as an environment variable for the other (i.e., both ϕ_l and ϕ_r are non-trivial). In order to ensure that the interconnection is well-posed (i.e., the interconnected system avoids deadlocks), the environment variables should be partitioned into external and feedback parts. For subsystem S_l, external environment variables are G_L and A_L, while the feedback environment e_f is the status of contactor C_4. In order for the system to be well-posed, decisions made by the controller for S_l at step t must use the value of C_4 at the previous step $t-1$.

Realizability is more difficult to achieve for the bidirectional case due to the issue of well-posedness. In order to successfully synthesize controllers for each subsystem, the following guarantees/assumptions are imposed: For S_r, if neither G_R nor A_R is healthy, then bus B_2 is powered $\phi_r = \square\{G_R = 1 \vee A_R = 1 \vee B_2 = 1\}$. For S_l, if neither G_L nor A_L is healthy, then power will be delivered through C_4 $\phi_l = \square\{G_L = 0 \wedge A_L = 0 \rightarrow (C_4 = -1)\}$.

Because power must be able to be delivered to the other subsystem when needed, safety-critical buses are moved

to those buses nearest the interface (e.g., B_2 and B_3.) In order to enforce well-posedness (i.e., to avoid deadlock), specifications for the controller for S_l involving C_4 are defined with additional next operators to implement a shift in time step. For the bidirectional synthesis problem to be realizable, contactor delays are thus omitted in this problem formulation in order avoid conflicting specifications.

There are design trade-offs in synthesizing centralized versus distributed architectures. A centralized controller has complete knowledge of all components' statuses. It can anticipate the behavior of the entire environment, and thus control protocols can be less conservative (e.g., longer delays in contactor switching times). For large-scale systems, though, distributed synthesis can be solved faster (due to the smaller number of components) and are thus more scalable. Additional refinements, however, are required at the interfaces. These refinements include more conservative contactor and bus configurations, (e.g, buses at the interface need to be powered more often). For the bidirectional distributed case in which refinements ϕ_l and ϕ_r are needed, well-posedness conditions further restrict the system. Contactor delays are no longer possible, and additional specifications are imposed on all components along the interfaces.

VII. Conclusion

This paper demonstrates how text-based specifications can be translated into a temporal logic specification language and used to automatically synthesize a control protocol for an electric power system on a more-electric aircraft. The resulting controller is guaranteed, by construction, to satisfy the desired properties even in the presence of an adversary (i.e., changes in the environment.) We synthesized a centralized controller, and then refined the interface specifications for distributed control architectures. Distributed controllers are easier to synthesize due to fewer components, but are more conservative with respect to power usage due to lack of information of the entire system.

ACKNOWLEDGMENTS

This work was supported in part by the Multiscale Systems Center and the Boeing Corporation. The authors wish to acknowledge Rich Poisson from Hamilton-Sundstrand and Necmiye ozay for their helpful discussions.

References

[1] J.S. Cloyd, "Status of United States Air Force's More Electric Aircraft Initiative," in *IEEE AES Systems Magazine*, pp. 17-22, April, 1998.

[2] L. Faleiro, "Initial research towards a more electrical aircraft," in *More Electrical Aircraft Conference*, Royal Aeronautics Society, 2004.

[3] J. Bals, G. Hofer, A. Pfeiffer, and C. Schallert, "Virtual Iron Bird - a multidisciplinary modeling and simulation platform for new aircraft system architectures," in *German Aerospace Conference*, Germany, 2005.

[4] P. Krus, B. Johansson, and L. Austin, "Concept optimization of aircraft systems using scaling models," in *Recent Advances in Aerospace Actuation Systems and Components*, France, 2004.

[5] P. Krus and J. Nyman, "Complete aircraft system simulation for aircraft design - paradigms for modeling of complex systems," in *22nd Intl. Congress of Aeronautical Sciences*, UK, 2000.

[6] I. Moir and A. Seabridge, *Aircraft Systems: Mechanical, Electrical, and Avionics Subsystems Integration*. AIAA Education Series, 2001.

[7] R.G. Michalko, "Electrical starting, generation, conversion and distribution system architecture for a more electric vehicle," US Patent 7,439,634 B2, Oct. 21, 2008.

[8] A. Pnueli, "The temporal logic of programs," in *Proc. of the 18th Annual Symposium on the Foundations of Computer Science*, pp. 46-57. IEEE, 1977.

[9] C. Baier, and J.P. Katoen, *Principles of Model Checking* MIT press, 1999.

[10] A. Pnueli and R. Rosner, "Distributed reactive systems are hard to synthesize," in *Proc. 31st IEEE Symp. Found. of Comp. Sci.*, pp. 746-757, 1990.

[11] A. Pnueli, Y. Sa'ar, and L.D. Zuck, "JTLV: a framework for developing Verification Algorithms," in *CAV*, pp. 171-174, 2010.

[12] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R.M. Murray, "TuLiP: a software toolbox for receding horizon temporal logic planning," in *Intl. Conf. Hybrid Systems: Computation and Control*, 2011.

[13] N. Ozay, U. Topcu, and R.M. Murray, "Distributed power allocation for vehicle management systems," in *Conf. on Decision and Control*, 2011.

[14] J.A. Rosero, J.A. Ortega, E. Aldabas, L. Romeral,"Moving towards a more electric aircraft," in *IEEE Aerosp. Electron. Syst. Mag.*, vol. 22, no. 3, pp 3-9, March, 2007.

[15] T. Wongpiromsarn, U. Topcu, and R.M. Murray,"Formal synthesis of embedded control software for vehicle management systems," in *AIAA Infotech@Aerospace*, 2011.

[16] N. Ozay, U. Topcu, T. Wongpiromsarn, and R.M. Murray, "Distributed synthesis of control protocols for smart camera networks," in *Intl. Conf. on Cyber-Physical Syst.*, 2011.

[17] T. Wongpiromsarn, "Formal methods for design and verification of embedded control systems: application to an autonomous vehicle," Ph.D. dissertation, Dept. Control and Dynamical Syst., California Inst. of Tech., Pasadena, CA, 2010.

[18] A. Pnueli, "Applications of temporal logic to the specification and verification of reactive systems: a survey of current trends," in *Current Trends in Concurrency. Overviews and Tutorials*, pp. 510-584, 1986.

[19] N. Piterman, A. Pneuli, and Y. Sa'ar, "Synthesis of reactive(1) designs," in *Verification, Model Checking and Abstract Interpretation*, vol. 3855, pp. 364-380. Springer-Verlag, 2006.

[20] R. Bloem, S. Galler, N. Piterman, A. Pnueli, and M. Weiglhofer, "Automatic hardware synthesis from specifications: a case study," in *In Design, Automation and Test in Europe*, 2007.

[21] T. Wongpiromsarn, U. Topcu, and R.M. Murray, "Receding horizon temporal logic planning," in *IEEE Trans. Autom. Control*, 2010.

[22] R. Bloem, et al., "Synthesis of reactive(1) designs," in *Journal of Computer and Systems Science*, 2011.

[23] S. Sohail and F. Somenzi, "Safety first, a two-stage algorithm for LTL games" in *Formal Methods in Computer-Aided Design*, pp. 77-84, 2009.

[24] M. Mukund, "From global specifications to distributed implementations," in *Synthesis and control of discrete event systems*, pp. 19-34, Kluwer, 2002.

[25] E. Filiot, N. Jin, and J.F. Raskin, "Compositional algorithms for LTL synthesis," in *Automated Technology for Verification and Analysis*, pp. 112-127, 2010.

[26] P. Madhusudan and P. Thiagarajan, "Distributed controller synthesis for local specifications," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, F. Orejas, P. Spirakis, and J. van Leeuwen, eds. vol. 2076, pp. 396-407, Springer, Berlin, 2001.