Use of Controls Approaches for Verification, Validation and Certification of Distributed Mission Management Controls for Unmanned Systems

AFRL S5, 2017

Kayin Cannon (NGC), Prakash Sarathy (NGC), Irene Gregory (NASA LaRC), Evangelos Theodorou (GT)
<table>
<thead>
<tr>
<th>Acronyms/Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>• BBA – Behavior Bounded Assurance</td>
</tr>
<tr>
<td>• FMEA – Failure Mode and Effects Analysis</td>
</tr>
<tr>
<td>• FMECA – Failure Mode, Effects, and Criticality Analysis</td>
</tr>
<tr>
<td>• FPM – Flight Performance Model</td>
</tr>
<tr>
<td>• GNC – Guidance Navigation Control</td>
</tr>
<tr>
<td>• GRA – Ground Rules &amp; Assumption</td>
</tr>
<tr>
<td>• IL – Inner Loop</td>
</tr>
<tr>
<td>• LHP – Left Half Plan</td>
</tr>
<tr>
<td>• MM – Mission Management</td>
</tr>
<tr>
<td>• MO – Mission Operator</td>
</tr>
<tr>
<td>• MPC – Model Predictive Control</td>
</tr>
<tr>
<td>• MPPI – Model Predictive Path Integral</td>
</tr>
<tr>
<td>• OCP – Optimal Control Problem</td>
</tr>
<tr>
<td>• OL – Outer Loop</td>
</tr>
<tr>
<td>• OODA – Observe, Orient, Decide, and Act</td>
</tr>
<tr>
<td>• RL – Reinforcement Learning</td>
</tr>
<tr>
<td>• SA – Situational Awareness</td>
</tr>
<tr>
<td>• UxS – Unmanned x (ground, air, or space) System</td>
</tr>
<tr>
<td>• UxV – Unmanned x (ground, air, or space) vehicle</td>
</tr>
<tr>
<td>• VMS – Vehicle Management System</td>
</tr>
</tbody>
</table>
Certification Challenges

- Fundamental problem
  - Input Space $\otimes$ Decision Space $\otimes$ Output Space $\Rightarrow$ Infeasible Test Size
  - Need alternative to comprehensive testing of operational envelopes

- Application vertical specific issues
  - Unmanned Aerial Vehicle (UAV)
    - Military weapon systems [Safety, Mixed Criticality, Security]
    - Commercial aircraft [GrossWt $\propto$ SafetyReq]
  - Unmanned Ground Vehicle (UGV)
    - Military weapon systems
    - Commercial automotive [Profit motive is primary]
    - Commercial heavy vehicle [Safety driven by FMEA/FMECA]
  - Unmanned Underwater Vehicle (UUV) / Unmanned Surface Vehicle (USV)

Certification needs to address combinatorial test case explosion
Safety Challenges for UxV: 1…2…3…

1. Why is this hard?
   - Consensus on the right _safe_ behavior
   - No integrated tools/framework to establish assurance
   - Doing more -> More autonomy -> More complexity

2. Why is this costly?
   - Current verification relies primarily on exhaustive testing ($$$)
   - Software complexity (combinatorial) \( \propto \) Testing cost ($$$$)
   - Input Space \( \otimes \) Decision Space \( \otimes \) Output Space \( \Rightarrow \) Infeasible test size

3. Why is this so murky?
   - No objective / functional framework for safety characterization
   - “Do no wrong” - Prove that UAV will avoid situations in which it cannot cope with or handle
   - Regulatory uncertainty further complicates these issues

Safety is hard, but should you care?
Safety Drivers: Lifecycle Approach

• What’s out there?
  – Requirements specification/analysis/validation tools
  – Formal Methods: Model checking, reachability sets, syntactic parsing, automated case generation …
  – Design for safety: Composability contracts, stability/safety bounds, oracles, safety controllers, monitors …
  – Safety guidance for implementation (MILSTD882/JSSSEH/AOP52…), simplex architectures (multicore), adaptive scheduling
  – Verification tool chains: Coverage tools

• What do we need?
  – Requirements/arguments/formalism
  – Design for safety certification guidance, models, software constructs
  – Build/implementation guidelines
  – Test/verification process, evidence collection
  – Making the certification case/licensure

A holistic approach to safety demands resiliency
• Alternative Assurance Paradigm
  – Define finite set of functional characteristics or traits that are of primary importance for UxS operations
  – Includes notions that are typically safety-related (failures of subsystems functions), but also mission failures (loss of survivability and vehicle integrity)
  – Define a means of establishing criteria and thresholds for each

• Candidate Implementation Methods
  – Develop verification methodology spanning the software lifecycle. This implies the evidence to be collected covers each of the phases of the software engineering lifecycle, namely; a) requirements, b) design, c) implementation, and d) verification
  – For specific software architectural constructs, identify the impact of using bounded assurance with respect to testing and verification methods
Steps in BBA Application

1. Decompose the system’s primary functions/services
2. Analyze each service for its potential importance/usefulness
3. Create cases that can fall within the category of each service’s effective value/possibility
4. For each case, list both implausible/undesired cases and relevant/desired cases
5. Create applicable bounds that can separate the desired from the undesired

Identify & set functional bounds on subset of safety critical behavior
BBA is a umbrella of different technical approaches

1. Temporal Behavior Assurance and Schedulability
2. Control Theoretic Approach to bound functional behavior
3. Algorithmic Approaches to bound software behavior
4. Certification Case Development
5. Software Architecture based behavior assurance
Overview Behavior Bounded Approach (BBA)

• BBA is an approach to show how the functional behavior of the system can be bounded

• Safety certification can be grouped into broad categories
  – Aircraft Structural
  – Flying/Handling Qualities

• Implicit assumption for safety certification of manned aircraft
  – Pilot has been properly trained
  – Pilot will behave in a safe and reliable fashion
UAV adaptation needs for safety and certification*

• As stated in the paper cited: a self-aware aircraft, spacecraft or system is one that is aware of its internal state, has situational awareness of its environment, can assess its capabilities currently and project them into the future, understands its mission objectives, and can make decisions under uncertainty regarding its ability to achieve its mission objectives.

• A challenge for self-aware or autonomous systems to get safety certification is the ability to show that it will operate safely as it adapts to changes in the environment, considering it will be intractable to test all possible cases ahead of time.

• L1 adaptive control has been used in safety critical systems to show system safety by expanding the robust performance envelop to mitigate uncertainty and disturbance effects on stability.

• In addition, L1 adaptive control provides precise trajectory following and guarantees graceful degradation to performance under failure or unforeseen environment disturbances.

• A potential scheme to explore is to use L1 adaptive controls for the inner loop to support safety and trajectory following.

Key Challenges

• Overarching Technical Questions:
  1. Is it feasible to model mission management logic as a Outer-Inner control loop construct?
  2. If so, are the loops linear or linearizable?
  3. Is this representation amenable to control theory constructs vis-à-vis behavior bounding?
  4. Is there a way to measure stability overall in the Inner and Outer loops?
  5. What approaches are used to show the stability of the overall system?

• Certification Relevant Questions:
  1. Is it feasible to compare Outer-Inner loop representations of Mission Management with that of VMS (GNC) designs w.r.t. certification process?
  2. If a direct comparison is feasible, then is there any way to extend or extrapolate the “accepted process” used for GNC to apply it for mission management?
Variable Definition Exemplars

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
<th>Exemplars</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z_{\text{ref}}$</td>
<td>Desired Goal/Set Point</td>
<td>Desired waypoint</td>
</tr>
<tr>
<td>$y$</td>
<td>Observed Vehicle State</td>
<td>Loc, alt, heading</td>
</tr>
<tr>
<td>error$<em>{OL}$, error$</em>{IL}$</td>
<td>Error variable</td>
<td>Difference b/w desired &amp; actual</td>
</tr>
<tr>
<td>$u_{OL}$</td>
<td>Update (Mission Plan)</td>
<td>TOI</td>
</tr>
<tr>
<td>$x_{OL}$</td>
<td>Commanded Variable (OL)</td>
<td>Desired waypt/destination</td>
</tr>
<tr>
<td>$u_{IL}$</td>
<td>Controlled Variable (IL)</td>
<td>Actuator Controls</td>
</tr>
</tbody>
</table>
Treating Mission Management as a Control System

• The objective of Mission Management (MM) is to successfully accomplish a mission which includes flying safely and accomplishing mission objectives in a uncertain dynamic environment

• Modeling MM as an Outer Loop that is responsible for determining reference input (set point) to feed the Inner Loop

• For example, in a mission that involves the use of a sensor subsystem, one could characterize the OODA loop for this Mission Management task as:

  • Output ‘y’ : The Inner Loop will report sensor malfunction to the Outer Loop

  • Error ‘error’ : The Outer Loop will take as an input, the error between output (from sensor) and reference input (desired)
    – This error the Outer Loop uses to assess the impact it has on its current mission plan
    – The Outer Loop determines the meaning and importance of the error relative to the current mission objectives and priority
    – The Outer Loop will predict how the system will behave in the future, given this error in determining if the mission can still be achieved
    – The Outer Loop will decide if the mission can continue or cannot based on its knowledge of its system state and environment, current mission goals and its ability to accomplish the mission safely

  • Reference Input ‘z_{ref}’ : if the choice is to continue the mission, then either the Outer Loop will update the mission plan or keep the current plan, and if the choice is to Return To Base (RTB), then RTB plan will be generated.
    – The new mission plan will be a reference input for the Inner Loop to follow. The Outer Loop will determine the desired end system and the Inner Loop will attempt to carry out that desired state (i.e., error is zero)
Outer-Inner Loop Control (OIC) Characteristics

- **Characteristics of Inner Loop**
  - Faster Data Rates
  - Stabilization
  - Robustness
  - Command Following
  - Disturbance & Noise Rejection
  - Can be linearized around a set of operating conditions

- **Characteristics of Outer Loop**
  - Slower Data Rates
  - Multi-level with complex coupling
  - Discrete/Event or Continuous Driven
  - Non-linear
  - Input space may not be fully known ahead of time
  - Real-time decision making framework inclusive of safety and flight, focused on managing the mission

- **Strong well-proven control-theoretical formulation to support bounded behavior for safety analysis**
MMCS Building Blocks

A: Vehicle Path Planning

- OL: Selects destination based on current tasking priorities
- IL: Given a destination, attempts to navigate to destination.
- Output of OL becomes the set-point of the IL

B: Sensor Task Planning

- OL: Select Target Location to be imaged
- IL: Steer sensor to target aim point
- Output of OL becomes the set-point of the IL
MMCS Use Case A: Vehicle Route Planning

• Inner Loop (IL) – VMS
  – GNC function to steer the vehicle to desired position
  – Controlled Variables: Aircraft State
  – Observed Variables: Aircraft State
  – Feedback Function(s): Difference between the aircraft state and desired position (i.e., error)
  – Stability Criteria / Invariants: Places the poles in LHP to satisfy performance specifications

• Outer Loop (OL) – Mission Plan Manager
  – Mission objective driven function to maximize effectiveness across all tasking. Subset for a single tasking, maximize tasking objective metrics (Time to Target, Fuel Economy, Comms Coverage, Survivability etc.)
  – Controlled Variables: Mission Plan (= Waypoints/Target)
  – Observed Variables: Progress to Plan
  – Feedback Function(s): Mission Assessment of Plan Progress to Current Need(s)
  – Stability Criteria / Invariants: Stochastic description available
Use Case A: Vehicle Route Planning (contd.)

- GRA: QuasiSteady -> OOD sub-loop is constant, No external MO guidance/retasking

- Combine \{O-O-D\} as a single block that generates the mission plan update

- \{Act\} is the direction (updated plan) provided to the next level subsystem (IL: VMS)

- VMS assesses the desired position(s) w.r.t current state and generates a candidate trajectory for the vehicle, subject to FPM, vehicle health state and SA of external conditions. Computes actuator positions and drives control surfaces.
BBA as applied to Mission Management

• The decision space for MM is not necessary completely described by physical variables that are used in typical control theory. For instance,
  – Which mission objective to do first, based on the current state of the system, environment, priority of mission objectives
  – A subsystem failure, should it continue its mission or return to be based

• To make these decisions require that the system is able to assess its current situation (both the environment and its own capability), impact on mission plan, current mission objectives, and ability to predict weather replan is needed

• It is this ability to dynamically adjust its mission plan that makes the system not always predictable, which make it difficult to get certified

• MM decision space expands across more than the flight envelope, it also deals with the mission itself

• MM decision making can be bounded
  – Some of the decisions can be bounded by applying doctrine
  – At any instance of time MM will only give commands to VMS that are within its current flight performance capability
  – Apply MPC to show that constraints of the decision variables will remain bounded

• One approach can use predictive techniques to ensure that decision capabilities will remain bounded in some prescribed region
Model Predictive Control Methods

• Model Predictive Path Integral (MPPI) control is a MPC technique that provides a mathematical basis for developing optimal control algorithms based on stochastic sampling of trajectories
  – MPPI is a sampling-based algorithm which can be optimized for general cost criteria

• Given an optimal distribution
  – Solve the minimization problem by moving the probability distribution induced by the controller towards optimal probability measure

• MPPI combined with Reinforcement Learning (RL) to maximize the benefits and minimize the weakness of each method
  – RL determines the optimal control input with little prior knowledge about the environment, however, RL is slow to converge
  – MPC determines the optimal control input quickly, but requires accurate models or the system dynamics and environment

Model Predictive Control (MPC) Approaches

- MPC is a control approach that deals with determining an optimal control for a system that must make decisions in an uncertain and dynamic environment.
- MPC problem is given the following:
  - Current state of process (i.e., dynamic model of the process)
  - History of past control moves
  - Optimization cost function $J$ over the receding horizon determine the optimal control input
- Solve the Optimal Control Problem (OCP)
  \[
  \min J = \sum_{\tau=t}^{t+T} l(x(\tau), u(\tau))
  \]
  subject to
  \[
  x(\tau + 1) = Ax(\tau) + Bu(\tau)
  \]
  \[
  x(t+T) = 0
  \]
  where
  - $x(t)$ is state
  - $u(t)$ is control input
  - $A$ is state dynamic matrix
  - $B$ is input matrix
  - $l(x(t), u(t))$ is cost function

- MPC algorithm implementation steps
  - At each time step, determine the optimal control input for the prediction horizon problem
  - Use the first value of optimal control input
  - At the next time step, update the system prediction based
  - Repeat

- Pros
  - Online optimization
  - Implements only the first optimal control move of receding horizon control problem
  - Optimized the current time step, while keeping future time steps in account
  - Anticipate future events and can take control actions as needed
  - Handle both equality and inequality constraints on controlled- and manipulated-variables
  - Targets are selected in real time given the current conditions

- Cons
  - Requires accurate model of the system dynamics and environment

Applying MPC to BBA

• MPC has the potential to provide a flexible framework to predict behavior with respect to bounds in a stochastic manner.

• MPC can also provide a means to apply bounding mechanisms to Mission Management software.

• MPC provides a logical extension of the OIC approach to MM and thus has the potential to provide a certification basis.
Summary and Work Ahead

• We have introduced a concept of BBA as a means of managing risk and uncertainty vis-à-vis safety and certification

• A fundamental aspect of BBA is the ability to characterize behavior and establish, monitor and enforce bounds

• This brief presented one thread of BBA, and showed how Mission Management could be formulated as a Outer-Inner Loop control system

• We proposed a means of combining OIC with MPC to provide a mechanism to implement BBA...

• We expect to be able to present some results including simulation data for a representative problem.
THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN