Cognitive Emotion Layer Architecture for Intelligent UAV Planning, Behavior and Control

Corey Ippolito, Greg Pisanich
QSS Group Inc
Computational Sciences Division
NASA Ames Research Center
Moffett Field, CA 94035
cippolito@mail.arc.nasa.gov

Larry A. Young
Army/NASA Rotorcraft Division
NASA Ames Research Center
Moffett Field, CA
layoung@mail.arc.nasa.gov

Abstract—Remote planetary exploration by autonomous vehicles in uncertain environments requires dynamic and highly adaptive decision making, behavior, and control mechanisms to maximize the chances of successful mission completion. We present in this paper an adaptive architecture for cognition, behavior and control of an autonomous unmanned aerial vehicle (UAV) Mars explorer called the Cognitive Emotion Layer (CEL) architecture that uses dynamical emotional response mechanisms to model explorer's response to continuous stimuli and provides adaptive decision making and control capabilities for the exploration platform.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. BACKGROUND 2
3. CEL ARCHITECTURE DEFINITIONS 3
4. LATERAL NAVIGATION SYSTEM DESIGN 6
5. SMART CAMERA SENSOR DESIGN 8
6. MEMORY AND DYNAMIC NETWORKS.............11
7. UAV NAVIGATION SYSTEM 13
8. CONCLUSION 15
9. REFERENCES 15
10. BIOGRAPHIES 16

1. INTRODUCTION

The goal of the Intelligent Aerial Vehicles (IAV) project at the NASA Ames Research Center is to investigate and develop novel reasoning and control technologies for remote planetary aerial exploration and scientific investigation. Aerial explorers enjoy many advantages over surface rovers, including a higher degree of mobility, instrument access to areas that cannot be traversed on the ground, and coverage of a larger area of the planetary surface [1][2][3]. Aerial Explorers would potentially be deployed in unexplored and largely uncertain environments where direct control is infeasible, requiring a high level of sophistication in autonomy for decision making and control that challenges state of the art techniques.

Bio-inspiration can be a powerful tool when applied to engineering problems, particularly the development of intelligent systems. In our project, we have applied biological inspiration to the development and demonstration of Mars-analog missions using terrestrial Unmanned Aerial Vehicles (UAVs) [4][6][7]. These missions were derived from individual biologically inspired behaviors. A natural progression was to investigate whether biological inspiration could lead to architectural or control structural definitions that could adaptively combine multiple behavioral definitions into a mission.

Several concepts for biologically inspired architectures were investigated, including an approach that combined emotional systems with holarchical structures [8]. Dynamical formulations and a layered emotional implementation were developed into a system that allows biologically inspired behaviors to be easily described at an atomic level. The combination of these atomic behaviors across multiple levels results in complex composite behaviors that would be difficult to define individually. The Cognitive Emotional Layer (CEL) architecture provides a single architecture that encompasses both the ability to define and implement high-level adaptive decision making as well as the lower-level stability and control of the aerial vehicle platform.

This paper introduces the Cognitive Emotional Layer architecture. We discus its relationship to emotional modeling and dynamical systems, provide a rigorous mathematical description of the architecture and use, then demonstrate how the CEL architecture was used in the design and implementation of three intelligent control structure applications: an adaptive UAV lateral control mechanism, a smart sensor implementation, and persistent memory mechanisms using dynamic networks. It concludes with an overview of a full UAV navigation system that is currently in development at NASA Ames using the CEL architecture and methodology.

[1] 0-7803-8870-4/05/$20.00© 2005 IEEE
[2] IEEEAC paper #1417, Version 2, Updated December 10, 2004

2. Background

Emotional Modeling and Artificial Intelligence

There are numerous initiatives to create emotion-based intelligent systems in the literature; previous work demonstrated the feasibility of emotional controllers for higher-level cognition and decision-making, typically geared towards emotional behavior and mimicking human responses. Modeling an emotional system for all aspects of vehicle control, however, is an approach whose utility and implementation feasibility remain to be demonstrated.

The development of an emotion-based system requires formalization of a consistent model for emotions that is practical and machine-implementable. The researcher must appreciate the fundamental shortcomings of this endeavor; the characterizations will yield a functional description of the emotional system, defining and assigning quantifiable values to a complex system that is notoriously hard to define and near impossible to quantify. This makes argument on the utility of emotion based systems difficult; variant and often conflicting classifications, definitions, and implementations yield fundamentally different results, and aspects of one approach will not necessary be reflected in another.

Despite the fact that many different models for emotional simulation have successfully been implemented, there is general recognition that biological components and mechanisms that evoke emotional reactions in animals to environmental and cognitive stimuli are not well understood [10]. Further, current capabilities of computers to process data might still be well short of that necessary to simulate such a complete model. The best approaches adapt cognitive models of emotion from the research in neuroscience, physiology, psychology, and even philosophy ([12]-[19]), capturing or simulating specific classifications from those models that emulate the expected behavior demonstrated by emotional organisms.

In one such approach, the OZ project at CMU [13] adds a higher-level emotion based cognitive layer (the Em module) above an unemotional lower level to close the perceive-think-react loop. This model is loosely based on cognitive models of humans described in [14]. Ventura [12] contrasts this approach of placing a high level emotional layer above a lower level unemotional layer with a *functional approach* that is constructed emotion-based throughout. An example of this approach is given in [15], where a society of 'emotion proto-specialist' agents, each associated with a particular emotion, contributes to the emergent emotional behavior in a particular way.

In [16], a two-layered system is presented based on a dualism found in several theories on human cognition, including the Canon-Bard theory and Papez circuit theory [10]; the system has two layers for processing stimuli input: a slower cognitive processor which extracts cognitive features of the stimulus to form a *generalized image model* (for instance, the image of a zebra can be evaluated as an animal with four legs attached to a body, stripped coloring, etc.), and a perceptual processor for more basic and immediate instincts that produce a *vector of desirability* (e.g., a lion's perception of a zebra triggering its predatory instincts). The generalized image model is a database of information that might be rich, structured, divisible, and complex. The vector of desirability contains information that is simple, indivisible, and implemented as an ordered list of values relating to certain characterizations of the object, such as is it positive or negative, desirable or avoidable, edible or inedible, etc. The dual representations are used for reasoning purposes, where fast reasoning or reflexive actions can use the desirability vector, while slower cognition can access the generalized image model. A set of complementary mechanisms use data from one model to adjust the other.

McCauley in [17] presents a system based on the psychological theory called 'pandemonium theory' [18][19]. In this system, each emotion is represented by an agent called a codelet. The analogy of an arena is used, with stands, a playing field, and sub-arena. A multitude of codelets populate the arena. Codelets on the playing field are active, doing whatever they were designed to do, while codelets in the stand watch the activities of the codelets on the playing field, waiting for something to excite them. The level of excitation of a codelet in the stand is associated with how loud the codelets yell, which also excites other codelets. When excited to a certain level, a codelet will activate and move to the playing field to perform its action, which will in turn excite other codelets in the stands. Codelet actions are linked to other codelets with certain gains like links in a neural network. When entering the playing field, the sub-arena creates input and output associations between the entering codelet and the currently active codelets. This sub-arena performs the actual input and output functions of the system. The current goal context of the system emerges from the active codelets on the playing field. High-level concept codelets may remain on the playing field for quite a long time, influencing the actions of the whole agent for that time. Multiple goal contexts might be competing or cooperating to accomplish their tasks.

Dynamical Systems Approach

The conceptual framework used in the research of cognition has profound effects on the consequent formulations, approaches taken, and, generally, the research performed in artificial intelligence. Recent approaches in the field of Cognitive Science have applied dynamical system theoretic techniques to model machine cognition that focus on dynamical cognitive structures in continuous interaction with the environment. The term *dynamical hypothesis* has been applied to this approach [21][22][23], a companion, arguably, to the traditional *computational hypotheses* paradigm. Symbolist models are based on the venerable presupposition that underlying cognition is the purely formal manipulation of quasi-linguistic symbolic representation by syntactic rules [1][21]. Connectionist models have also gained widespread usage, where control is distributed among primitive elements arranged in a network, often processed in parallel, where knowledge is distributed in the form of patterns of connectivity among the elements. The dynamical approach represents a third approach, where cognitive agents are modeled as dynamical systems that evolve over time governed by nonlinear differential equations. There is a growing number of architectures being developed based on a dynamical approaches in the literature. In fact, many of the emotional systems described previously implicitly use equations to propagate system states that are governed by linear differential equations in their derivatives.

The philosophical similarities, differences, rational, advantages, and limitations of dynamical approaches have been well addressed in the literature and is a subject of ongoing – often spirited – debate [24][25][26][27][28]. The authors do not intend to engage this debate here; the merits, advantages, and disadvantages of each system have been well documented and can be found in the literature. Rather, we present a formal system that is dynamical in nature, and describe its successes and limitations. Although a formulation and practical application framework and methodology is presented for endowing machines with adaptive and continuous behavioral control and evaluate its results from an entirely dynamical perspective, the authors suggest this kind of framework would find most successful application in conjunction with traditional symbolist or connectionist models. Part of the purpose of the CEL architecture is to probe dynamical formulations as a basis for practical tools in artificial intelligence, evaluating its actual strengths in application beyond generalized abstract arguments. Another driving goal in the formulation is to establish a strong connection between the dynamical hypothesis and the elegant and powerful fields of dynamical systems and control systems theory by establishing a general mathematical and computational framework conducive to traditional methodological analysis; this connection - possibly the most vital argument in favor of the dynamical hypothesis – provides a science and tools that are tested and mature.

The Cognitive Emotion Layer Architecture

This paper introduces a software architecture and formulation that postulates dynamical structural formulae for the creation of composable mental networks for adaptive decision making and cognition as well as for low level – fast, simple, 'close to the hardware' – stability and control of the vehicle platform. This architecture is an attempt to extend the formulations of previous emotional software systems, providing a link to dynamical system and control theory. For our purposes we define an emotional state in very broad terms as a reactive time-varying cognitive variable that responds to stimuli and mediates between stimulus and a response. Emotional states are modeled as part of transformational networks that are designed to drive the system to desirable states. At the same time, we avoid associating this definition of emotion with that of human interpreted qualitative states such as happiness, anger, hunger, etc. Rather the emotional mechanisms are designed for utility in autonomous exploration without human analogue. The network structures can be tuned to make explorers more aggressive in their search patterns, less likely to cast doubts on their previously held assumptions of the environment, more attuned to perceptual stimuli, or place higher weight on 'introspective' loops where emotional stimuli feedback on themselves. Higher level emotional states are achieved through layering and compositing of networks, though the exact nature of these states is specific to each system and application, without an attempt to force parallels between the machine states and subjective high-level human states.

3. CEL Architecture Definitions

The Cognitive Emotion Layer architecture provides a structure for implementing emotion based reasoning, intelligent maneuvering, decision-making, behavior selection, and control of autonomous unmanned aerial vehicles. The architecture allows emotional constructs, CELs, to be layered to construct cognitive systems. Conceptually, as shown in Figure 1, the CEL processor hardware transforms stimulus inputs from hardware sensors, timers, etc, and produces output for driving the actuators and manipulators on the platform.

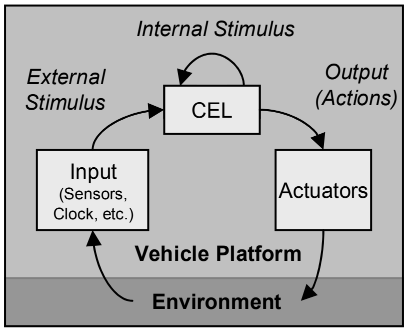


Figure 1 – UAV Hardware Diagram

A high-level component diagram for a CEL-based cognitive system for exploration is shown in Figure 2. CEL networks can be reduced to simple graphs that can then be restructured along traditional look-think-act boundaries. However, the approach we define for composition and layering tends to create components that straddle these broad classifications; a responsibility driven interpretation of each component is instead offered as a means of understanding and designing CEL networks.

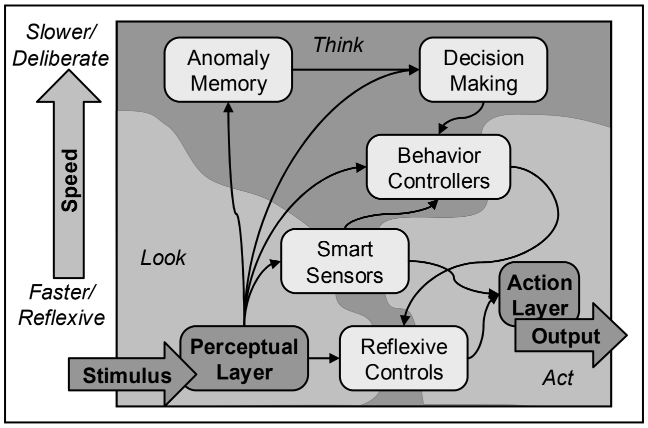


Figure 2 – CEL Explorer System
(Classic Look/Think/Act Divisions Shown)

Emotional vertices, also called codelets for historical reasons, are the basic transformation elements in the CEL architecture and contain the following properties:

1. a time-varying input vector $\mathbf{u}(t)\in\Re^m$
2. an internal time-varying state vector $\mathbf{x}(t)\in\Re^n$
3. a time-varying output vector $y(t)\in\Re^l$
4. internal parameters vector $\mathbf{p}\in\Re^p$
5. a propagation relation τ which defines how the four parameters above behave in time

An **emotional vertex** can be define as a set $\mathbf{V}=(\mathbf{u},\mathbf{x},\mathbf{y},\mathbf{p},\tau)$, where $\mathbf{u}\in\Re^m$, $\mathbf{x}\in\Re^n$, $y(t)\in\Re^l$, and τ is the propagation relation. To **valuate** an emotional vertex is to propagate the vertex forward in time by a discrete time step. The main processing step in updating a CEL network is the **valuation process**, where each node in the graph is valuated in a certain order that maintains the coherency and consistency of the model.

An e**dge** $\mathbf{E}=(s,t,w)$ in an emotional layer network transports a value from one vertex to another; edges can be considered for computation reasons to be an instance of a vertex $E\subset V$ where $E=V(s,\varnothing,t,w,\tau)$ where $s,t\in\Re$ are the tail and head variables respectively, $w\in\Re$ is the **edge weight**, and $\tau(s,w)=s*w$. We use the operators t[e] and s[e] to return the tail and head vertices ($t[e],s[e]\in V$) of an edge e.

A **cognitive emotional layer** is a sub-network of a complete cognitive network, defined as a set of vertices and edges $\mathbf{L}=(\mathbf{V},\mathbf{E})$. Note that **L** is not necessarily a graph, in that for $e\in\mathbf{E}[\mathbf{L}]$, where $\mathbf{E}[\mathbf{L}]=\mathbf{E}$, there is no guarantee that $s[e],t[e]\in\mathbf{V}[\mathbf{L}]$, where $\mathbf{V}[\mathbf{L}]=\mathbf{V}$. The **edge-in set** $\mathbf{I}[\boldsymbol{l}]$ of a layer $\boldsymbol{l}\in\mathbf{L}$ is defined as the set of edges where $\mathbf{I}[\boldsymbol{l}]\equiv\{e:s[e]\notin\mathbf{V}[\boldsymbol{l}],\ t[e]\in\mathbf{V}[\boldsymbol{l}]\ \forall e\in\mathbf{E}\}$, or the set of all edges that point to vertices in a layer $\boldsymbol{l}$ that started from vertices outside of that layer. The **edge out-set** $\mathbf{O}[\boldsymbol{l}]$ of a layer $\boldsymbol{l}\in\mathbf{L}$ is similarly defined as $\mathbf{O}[\boldsymbol{l}]\equiv\{e:s[e]\in\mathbf{V}[\boldsymbol{l}],\ t[e]\notin\mathbf{V}[\boldsymbol{l}]\ \forall e\in E[S]\}$, or the set of all edges that start from vertices inside of the layer and point to vertices out of the layer.

If a layer $\boldsymbol{l}\in\mathbf{L}$ contains a non-empty $\mathbf{I}[\boldsymbol{l}]$ or $\mathbf{O}[\boldsymbol{l}]$ set, the network is considered to be **incomplete** and is not instantiable due to the lack of connections. However, tuning large networks with multiple interdependent variables is often difficult, so layers are often instantiated independent of a full cognitive network in order to analyze and tested the independent function of the sub-system under controlled input/output conditions. The incomplete layer can be *completed independently* by creating support layers and nodes to terminate the edges in $\mathbf{I}[\boldsymbol{l}]$ or $\mathbf{O}[\boldsymbol{l}]$, creating an instantiable and complete system for debugging and tuning purposes.

A **composition** is defined as set of layers $\mathbf{C}=(\mathbf{L}_0, \mathbf{L}_1, \ldots)$ where any edge $e\in\mathbf{E}[\mathbf{C}]$, $\mathbf{E}[\mathbf{C}]=\{\mathbf{E}[\mathbf{L}_0],\mathbf{E}[\mathbf{L}_1],\ldots\}$, has endpoint vertices s[e] and t[e] that may not be an element of $\mathbf{V}[\mathbf{C}]=\{V[L_0],V[L_1],\ldots\}$; i.e., like layers, compositions can be incomplete.

A useful approach to formulating complete CEL networks is to pose the problem as a search problem with an initially empty composition, and a complete CEL system is constructed by **compositing**, or growing the composition set **C** with the addition of layers, until a final goal state is achieved. In fact, this formulated **compositing problem** can allow systems to self-assemble a particular network solution using classic strategies such as A* or genetic algorithms.

A **CEL System** is a *complete* network composition of CEL sub-networks, and defined as $\mathbf{S}=(\mathbf{L}_0, \mathbf{L}_1, \ldots)$. Here, complete means edges in a system reference vertices contained within one of the L_i layers of the system. Let $\mathbf{E}[\mathbf{S}]=(\mathbf{E}[\mathbf{L}_0]\cup\mathbf{E}[\mathbf{L}_1]\cup\ldots)$, and $\mathbf{V}[\mathbf{S}]=(\mathbf{V}[\mathbf{L}_0]\cup\mathbf{V}[\mathbf{L}_1]\cup\ldots)$. Then a CEL system S is a *complete* composition, where for any $e\in\mathbf{E}[\mathbf{S}]$, $(s[\mathbf{e}],t[\mathbf{e}])\in\mathbf{V}[\mathbf{S}]$. A CEL system is a graph, and is given as $\mathbf{G}[S]=(\mathbf{V}[S],\mathbf{E}[S])$.

Consider a complete CEL system **S**. The **influence graph** of a vertex $\mathbf{v}\in\mathbf{V}[\mathbf{S}]$ is the subgraph of **S** that contains all edges and vertices that are involved in the valuation of **v**. Let $\mathbf{V}'=\{\mathbf{u}\in\mathbf{V}[\mathbf{S}]:\exists\mathbf{u}\sim\mathbf{v}\}$, then the influence graph of the vertex **v**, **H**[**v**], is defined as the subgraph of **G**[S] induced by **V**', or $\mathbf{H}[\mathbf{v}]=\{\mathbf{V}',\mathbf{E}'\}$ where $\mathbf{E}'=\{(\mathbf{u},\mathbf{v})\in\mathbf{E}[\mathbf{S}]:\mathbf{u},\mathbf{v}\in\mathbf{V}'$ and $w[(\mathbf{u},\mathbf{v})]\neq0\}$. Example influence graphs are shown in Figure 3. An influence graph **H**[**v**] may by acyclic, cyclic where a cycle includes the vector **v**, or cyclic where **v** is not contained in a cycle. For any vertex **v**, **H**[**v**] has the following properties:

(1) **H**[**v**] is unique

(2) **H**[**v**] is a subgraph (inclusive) of the complete component of G[S] that contains v

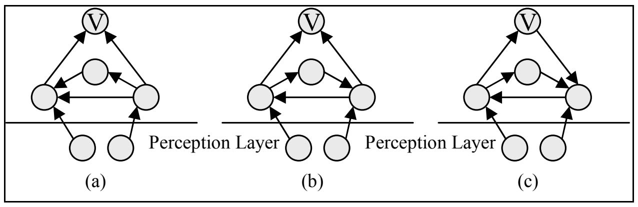


Figure 3 - Influence Graphs of Vertex V
(a) Acyclic Influence Graph, (b) Cyclic, (c) Cyclic in V

Influence graphs are used for analysis of network behavior by defining subgraphs that capture how a disturbance propagates through the system, a first step in pruning the unimportant information when tracing behavior in a system. Influence graphs also are used when determining node ordering for processing during the vertex valuation process.

Vertex Types and Classes

Much of the promise of this architecture is ability to create reusable emotion vertex definitions that can be used as primitives to facilitate design and implementation of new CEL networks, as well as the ability to design reusable sub-networks that can be composited to form networks that are more capable and complex. The definition of the emotional vertex given is very broad, generically encompassing a large class of possible transformations. The following vertex classifications and definitions were created to provide a concrete set of reusable primitives in creating CEL networks. Many of these definitions are similar to the primitive definitions for emotional systems in the literature; however, the formulations given here tend towards using differential equations to define state variable behavior as opposed to explicit relationships defining system state behavior over time.

An analytical node is a simple processing node that performs analytical transformations of the raw data. Many of the CEL systems the authors have developed contain an analytical layer; a CEL layer that processes and filters the raw sensor data into more usable signals for other layers. Analytical nodes include nodes such as Kalman filters and feedback controllers.

The following definition defines a class of analytical nodes that acts as a proportional-integral-differential (PID) controller vertex for use in networks were an error signal can be minimized through PID feedback. The PID codelet's input vector uPID=[e e0]T has two elements: an error signal input e and the desired error e0. The propagation function τPID is given by

$$\tau_{pid}(\mathbf{u},t) = k_p e(\mathbf{u}) + k_i \int_0^t e(\mathbf{u}) + k_d \tfrac{d}{dt} e(\mathbf{u}) \qquad (1)$$

where $e(\mathbf{u})=u_2-u_1$. In the CEL library's particular implementation, the PID vertex contains the state vector $\mathbf{x}_{pid}=[i_{state},e_{last}]$, where i_{state} is the current integrator error term, and e_{last} is the the previous error value at the last discrete time step (for first-order derivative approximation). The output vector $\mathbf{y}\in\Re$ is the output of the PID controller. The class of PID vertices is defined as

$$V_{pid} = \left(u_{pid}, x_{pid}, y_{pid}, \lfloor k_p, k_i, k_d \rfloor, \tau_{pid}\right) \qquad (2)$$

Anxiety nodes are a classification of vertices where a particular value or set of values in the output or internal state are identified as **anxiety parameters**. The purpose of anxiety nodes is to process stimuli into an anxiety value representing an emotional attraction or dissatisfaction with the current explorer state. An **outlet behavior control** must be defined in the network or in the anxiety node itself; an outlet behavior control is a mechanism designed into the network which identifies a set of variables through which the anxiety values are *controllable*.

A **concern node** is an instance of an anxiety node $\mathbf{V}_c=(u,\varnothing,y,\{k_1,k_2\},\tau)$ where $u,y,k_1,k_2\in\Re$ and the anxiety parameter's behavior is governed by a first-order nonlinear differential equation of the form

$$\frac{dy}{dt} = k_1 u + k_2 u y \qquad (3)$$

Concern nodes are used to filter raw data signals into a continuous and differentiable form, or to accumulate signals into a signal parameter that often represents the 'concern' the system has some target phenomena. For instance, a concern node is defined to compute a single 'fuel usage concern' value that grows and shrinks as a function of the fuel consumption rate and battery power level rate of decline.

A **desire node** $\mathbf{V}_c = (u, \dot{y}, y, \{k_1, k_2, k_3\}, \tau)$ is an instance of an anxiety node where $u, \dot{y}, y, k_1, k_2, k_3 \in \Re$ and the anxiety parameter's behavior is governed by a second-order linear differential equation of the form

$$k_1 \frac{d^2 y}{dt^2} + k_2 \frac{dy}{dt} + k_3 = u \qquad (4)$$

Desire nodes are used to model desires and preferences, where 'forcing' variables, often the output of concern nodes, provide positive or negative influences on the desire. Desire nodes often are used for selection between a set of possible items, such as selecting a particular behavior, where each desire node in the selection represents a preference for the associated behavior. These nodes are often grouped into a **normalized desire group**, where the desire anxiety parameters are constrained so that the sum of the squares of the values is constant. This constraint adds non-linearity to the node's behavior.

4. Lateral Navigation System Design

The CEL architecture definitions were used to implement control structures that performed a number of low-level behaviors and would adaptively combine the behaviors in response to stimulus that sensors received from the environment. This section describes the basic approach for developing a system in the CEL architecture through a simple lateral navigation system example; this example includes definitions of atomic components used in the full autonomous navigation system currently in development at NASA Ames.

Terrain Avoidance Anxiety

The intelligent exploration of remote planetary surfaces such as Mars requires some heuristic be identified to help reduce total search space and enable the explorer to make more intelligent decisions during a mission. Consider exploration over a 2-dimensional landscape (the workspace), ignoring altitude, and searching for multiple discrete targets. Let a 2-dimensional continuously differentiable potential field be mapped onto the workspace which represents areas of expected probability that the goals may or may not be located in this region. This could be a search, for instance, for minerals expected to be found in a dried out riverbed, and the probability mapping is the actual terrain elevation. The heuristic dictates that the mineral is more likely to be found at lower elevations (the riverbeds) than higher elevations.

In the emotional CEL network, the continuous input of this probability can be regarded as negative stimulus into a perceptual sub-network dedicated to processing this topology. We design a network where this processed stimulus gets passed into a reflexive sub-network that implements the searching behavior, and a slower deliberate cognitive layer that makes higher level decisions. The reflexive emotional network will be responsible for implementing a behavior that will follow this heuristic. The cognitive network will be responsible for making higher level decisions about the quality and applicability of this heuristic.

Constructing complete systems in the CEL system can be accomplished using the following iterative process for designing new behaviors into a network:

(1) Define the new behavior.

(2) Identify design points for the system and diagram desired system behaviors and behavioral interactions.

(3) Design the network components to achieve the desired behaviors.

(4) Analyze the system to determine appropriate gain and parameter settings.

(5) Iterate for each additional behavior (repeat steps 1-4).

(6) Integrate into the system composition.

(7) Analyze paths in the graph through influence graphs and simplifying assumptions to eliminate edges to determine appropriate gain and parameters settings.

This process is used the following sections for each of the designs. The first step is to define the behavior, identify internal state variables and their desired classification (such as anxiety, desire, etc.), and then to design outlet behaviors, which are mechanisms that provide some way of controlling the internal states through behavior selection. A single 'terrain anxiety' node in the reflexive sub-network layer can do both, where anxiety increases as terrain increases and probability of finding targets decrease, and the behavioral outlet will be a simple greedy steepest descent strategy as a control mechanism for anxiety reduction.

The terrain anxiety node V_{ta} will compute a desired course that will attempt to follow terrain depressions. The desired heading can feed into a traditional autopilot system, or in this case, an emotional reflexive layer component. Consider the explorer in Figure 4 at position $\mathbf{P}=(x_p,y_p)$ with a heading ψ_p. Let the elevation of the ground be given by h(x,y) where h is continuous and differentiable, let the gradient vector be $\mathbf{S}(\mathbf{P})=(\delta h/\delta x_p,\delta h/\delta y_p)|_{xp,yp}$.

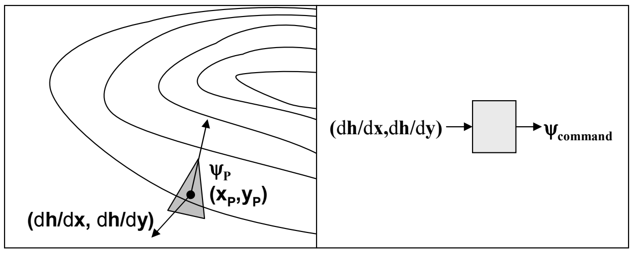


Figure 4. Terrain Anxiety

Let the input vector for the terrain anxiety emotional vertex be given by $\mathbf{u}_{ta}=[\mathbf{S}\ \mathbf{P}\ h]^T$. The output vector $\mathbf{y}_{ta}=[A, \psi_{ta}]$, where A is the anxiety magnitude, and ψ_{ta} is the direction that will decrease the anxiety. The heading command ψ_{ta} can be simply computed as the steepest gradient direction, $\psi_{ta}=\text{atan2}(\|\mathbf{S}(\mathbf{P})\|_x,\|\mathbf{S}(\mathbf{P})\|_y)$, where $\|\mathbf{S}(\mathbf{P})\|_x$ and $\|\mathbf{S}(\mathbf{P})\|_y$ represent the first and second element of the normalized 2D slope vector, respectively, and atan2 is the quadrant-aware arctan function. The intensity of the anxiety will increase as a linear combination of the gradient magnitude and the altitude, or $A=\text{CAP01}(k_1h+k_2|\mathbf{S}(\mathbf{P})|)$. The function CAP01(y) will return 0 if y<0, else 1 if y>1, else y. Then the propagation function is given by

$$\tau_{ta}=\begin{bmatrix}\text{CAP01}\left(k_1h+k_2|S(P)|\right)\\ \text{atan2}\left(\|S(P)\|_x,\|S(P)\|_y\right)\end{bmatrix} \tag{5}$$

The constants k_1 and k_2 will depend on the units used and how much influence the size of the slope and the altitude of the terrain have on the overall vertex anxiety intensity. Let $\mathbf{p}_{ta}=\{k_1,k_2\}$. Then the terrain avoidance anxiety can be given as

$$\mathbf{v}_{ta}=\{\mathbf{u}_{ta}, \varnothing, \mathbf{y}_{ta}, \mathbf{p}_{ta}=\{K_1,K_2\}, \tau_{ta}) \quad (6)$$

Track to Waypoint Anxiety

Consider an aircraft tracking from waypoint A to waypoint B as shown in Figure 5. An anxiety node $\mathbf{v}_{wp}$ will be formulated that will take this input and compute a desired heading angle. The node's anxiety parameter will increase as a function of the cross-track error. The output vector for this node is $\mathbf{y}_{wp}=[\psi,A]$.

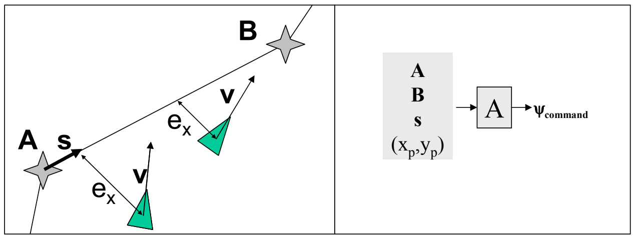


Figure 5. Waypoint Following Anxiety

Let A and B be the start and end points of the path, and **P** be the location of the aircraft. Let $s=\left\|\overline{AB}\right\|$, and the tangent vector $\hat{\mathbf{s}}$ from the point P to the line AB be given by

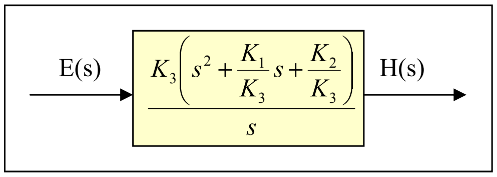
$$\hat{\mathbf{s}}=\left|(AB\times PB)\times AB\right| \quad (7)$$

$$\hat{\mathbf{s}}=\left\|\begin{bmatrix}-(AB\times PB)_z\cdot AB_y\\(AB\times PB)_z\cdot AB_x\end{bmatrix}\right\| \quad (8)$$

The (positive) cross-track error from the vehicle's position to the track-to line AB is given by

$$e_x=\left|PB\cdot\hat{\mathbf{s}}\right| \quad (9)$$

The cross-track error e_x is fed through a PID transformation block we will build into this anxiety node to determine a heading angle to take towards the track waypoint, as shown in Figure 6, where k_1, k_2, and k_3 are the PID gains.



E(s)
$\frac{K_3\left(s^2+\frac{K_1}{K_3}s+\frac{K_2}{K_3}\right)}{s}$
H(s)

Figure 6 - PID Transfer Function

The output signal from the PID block, t, is capped from 0 to 1, and used to compute a desired heading vector **v** and the commanded heading ψ_c, defining τ_{wp}.

$$\mathbf{v}=(t)\hat{\mathbf{s}}+(1-t)\mathbf{s} \quad (10)$$

$$\psi_c=\text{atan2}(\mathbf{v}_y,\mathbf{v}_x) \quad (11)$$

The anxiety parameter A will increase proportional to the distance between the aircraft and the track from waypoint A to B.

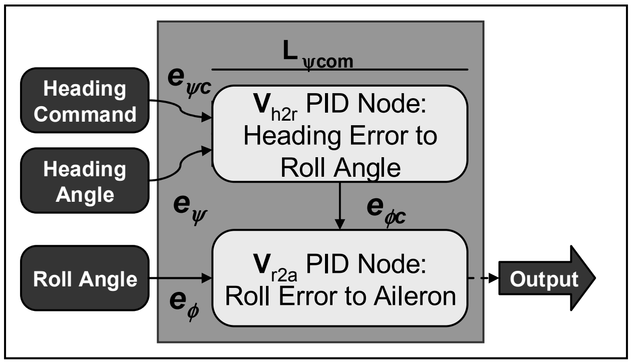
$$A=\text{CAP01}(k_4 * e_X) \quad (12)$$

Here, CAP01 is defined as before, and k_1 is a user defined gain. The vertex's internal parameter vector is $\mathbf{p}_{wp}=\{k_1,k_2,k_3,k_4\}$. The waypoint following anxiety vertex is then defined as

$$\mathbf{v}_{wp}=\{\mathbf{u}_{wp}, \mathbf{x}_{wp}, \mathbf{y}_{wp}, \mathbf{p}_{wp}, \tau_{wp}\} \quad (13)$$

Lateral Waypoint Navigation Layer

The PID vertex in Equation (2) is used prominently in reflexive autopilot sub-networks. The sub-network shown in Figure 7 is a simple lateral mode controller to command aileron deflection based on course heading error input signal.



L_ψcom
Heading Command
Heading Angle
Roll Angle
e_ψc
e_ψ
e_φ
e_φc
V_h2r PID Node: Heading Error to Roll Angle
V_r2a PID Node: Roll Error to Aileron
Output

Figure 7 - Lateral Heading Command Network

This lateral heading command layer is defined as L ψcom=(Vψcom,Eψcom), where Vψcom={vh2r, vr2a} contains two PID vertices vh2r,vr2a∈Vpid. The edge set Eψcom={eφ,eφc,eψ,eψc} contains edges eφ=(φ,u1[vr2a],1), eφc=(y[vh2r],u2[vr2a],1), eψc=(ψc,u2[vh2r],1), and eψ=(ψ,u1[vh2r],1), where φ, ψ, and ψc are vertices external to Lψcom.

Parameters in the PID vertices are a function of the vehicle platform's dynamics and are implemented as a function of the flight condition. In this case, the PID gains in p[vh2r] and p[vr2a] can be tuned for a vehicle in isolation by classical control system design techniques. Further this simple sub-network can be tested by instantiating the layer as a complete CEL system; this entails adding in perceptual and actuation layers, creating input and output vertices in these layers, and connected the three layers with edges between the appropriate vertices. This gives a component-wise method for creating and tuning simple sub-networks that will be composited into larger and more complex systems.

Lateral Mode Behavior Selection Composition

Reusable layer definitions like $L_{\psi com}$ can be composited to form more complex network structures. In Figure 8 a composition $\mathbf{C}_{LB}$ is shown. This is a step towards the complete system in Figure 2.

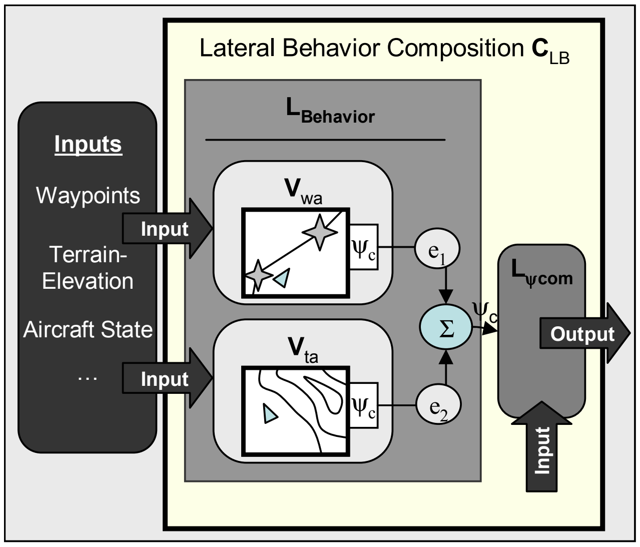


Figure 8 - Lateral Mode Behavior Composition

The behavior layer manages two anxiety nodes, $\mathbf{v}_{wp}$ (13) and $\mathbf{v}_{ta}$ (6). The blending between these two behaviors is specified by the edge weights on edges $\mathbf{e}_1$ and $\mathbf{e}_2$; an implicit constraint is

$$w[\mathbf{e}_1]+w[\mathbf{e}_2]=1 \tag{14}$$

The node labeled 'Σ' is a summation block that provides an output variable for other layers- in this case, $L_{\psi com}$, and the set of vertices in $L_{behavior}$ are $\mathbf{V}_b=\{\mathbf{v}_{wa},\mathbf{v}_{ta},\Sigma\}$, and the edge set is $\mathbf{E}_b=\{\mathbf{e}_1,\mathbf{e}_2,\ldots\}$, where the ellipses represent the implicit edges leading into V_{wa} and V_{ta} from nodes external to $L_{behavior}$ (similar to $L_{\psi com}$ in Figure 7). The composition and behavior layers are defined as

$$\mathbf{C}_{LB}=(\mathbf{L}_{behavior}, \mathbf{L}_{\psi com}) \tag{15}$$

$$\mathbf{L}_{behavior}=(\mathbf{V}_b,\mathbf{E}_b) \tag{16}$$

The edge weights $w[\mathbf{e}_1]$ and $w[\mathbf{e}_2]$ represent the balance between the desire to follow a waypoint and the desire to migrate towards lower terrain. This incomplete composition will composited with a higher level cognitive layer that will use these weights as control inputs.

Simulation Results

The CEL architecture formulation allows for layers to be created and tested independently, avoiding the problems of gain tuning in complex network structures with hundreds of interdependent parameters. By itself, the C_{LB} composition is a simple waypoint controller that is linearly combined with a steepest descent algorithm, but independently completing simple sub-networks is an important step for tuning parameters in a manageable manner. C_{LB} was completed independently and instantiated in a computer simulation of a Mars-class unmanned aerial vehicle explorer. A navigational computer database provides two waypoints on either side of a raised hill region. The simulation results are reported with three different edge weight desire ratios of w[e1]:w[e2] in equation (14).

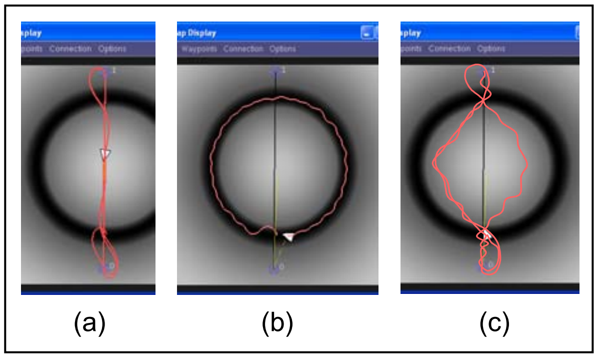


Figure 9 – Lateral Mode Behavior Sim Results

Figure 9 plots the resulting trajectory of the simulated aircraft's waypoint following behavior, simulated using a w[e1]:w[e2] ratio of (a) 1.0:0.0, (b) 0.0:1.0, and (c) 0.5:0.5.

5. Smart Camera Sensor Design

This section details the design of an emotional control system used to control the behavior of a camera sensor mounted on an intelligent unmanned aerial vehicle. The camera is mounted on a pan and tilt mechanism which allows two degrees of freedom: yaw rotation (ψ), and pitch rotation (θ). The two high level requirements for the camera control system are as follows: (1) control the position of the camera to search for targets on the ground terrain as the UAV is in flight; (2) provide command input to the aircraft to investigate areas when there is a reasonable expectation of finding a target; (3) provide an 'interest' metric in how strongly the sub-network feels that UAV should move towards the identified anomaly. The term 'reasonable expectation' is left up to the camera's emotional sub-network to define. The definition of the target is left intentionally vague, as the precise definition is mission dependent; for instance, the camera may be an infrared detector searching for infrared signatures on the ground, a standard video camera detecting motion, or detecting other signature patterns useful in exploration or surveillance. The interest metric is a complex function of different factors; we somewhat arbitrarily identified three reasonable assumptions about imaging anomalies that feeds into this metric: (1) should be imaged at a certain optimal distance, (2) imaged for a certain period of time, and (3) larger anomalies are more important that smaller anomalies. Of course, different assumptions would probably be made for real-world applications that would depend on the camera hardware and the characteristics of the real anomaly being imaged, and these assumptions would affect the network design.

The emotional camera control system (ECCS) layer is shown in relation to external hardware and an existing emotional flight control system (EFCS) layer in Figure 10. The arrows in this figure represent data flow between components.

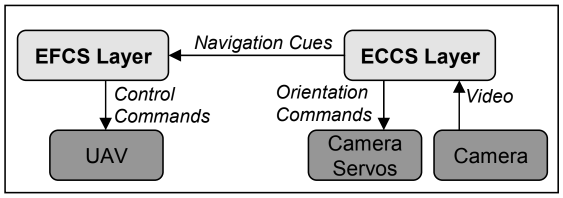


Figure 10 - EFCS, ECCS, and Hardware.

ECCS Design Using Iterative Network Construction

The ECCS provides pitch and yaw commands to the camera hardware (θ_c and ψ_c). To the EFCS component, the ECCS provide a parameterized confidence metric (σ) and a commanded heading (ψ_{Ac}), as shown in Figure 11. This component takes an input of the current time (t), and the current camera pitch and yaw angles (θ and ψ).

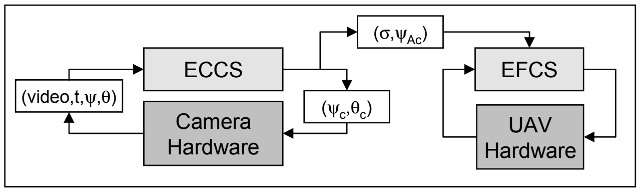


Figure 11. Input/Output of the ECCS

The network for the ECCS is designed using the iterative construction approach previously described.

Iteration 1: Sinusoidal Sweep and Neutral Point Behaviors

The first design iteration involves two basic behaviors. A sinusoidal sweep (SS) pattern commands the camera to move around the camera's state space in a sinusoidal pattern as shown in Figure 12. The neutral point (NP) command indicates a desired direction for the camera. The NP reflects the fact that areas of interest ahead of the aircraft can be navigated towards with much less energy than areas that are behind the aircraft. The camera control system will focus on areas ahead of the aircraft, and will lose interest in ground locations as the aircraft passes them.

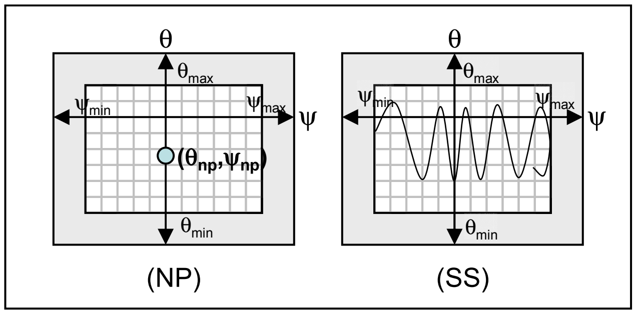


Figure 12. Neutral Position (NP) and Sinusoidal Sweep (SS) Behaviors

The implementation of CEL system analytical nodes for the SS behavior (ANSS) and NP behavior (ANNP) are trivial.

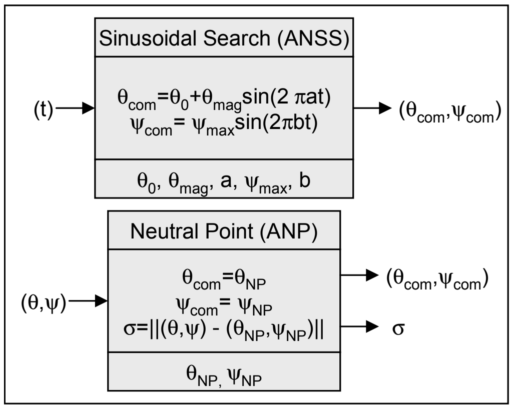


Figure 13 - Analytical Nodes for SS and NP

The system will maintain two desire nodes representing the desire to perform the SS behavior (DSS) and the desire to return to the NP (DNP). Both DSS and DNP will be placed in a normalized desire set (sum of the squares of the magnitudes of the DSS and DNP desires will always be equal to one). A 'not detecting anomaly' anxiety node (XNDA) will be used to implement displeasure at not finding the target material. An additional anxiety node will be used to implement displeasure at pointing away from the neutral position (XNP).

Consider the scenario with the camera initialized at the neutral position, DNP=1 and DSS=0 (i.e., the system when initialized is content to keep the camera pointing at the neutral position). The camera's sensors are not reporting any targets. The following graphs illustrate desired behavior. At t=0, anxiety levels are low and DNP is at max. As time progresses to t=A, XNDA increases due to the lack of target identification. XNDA should result in a downward pressure on the maximum desire, causing DNP to decrease and DSS to increase. As DSS increases, the effects of the ANNP commands will be more pronounced, eventually causing XNP to start rising. The camera will begin to move in sinusoidal sweeps of increasing magnitude about the neutral point. At time t=B, XNDA will saturate (in this scenario the system will not identify any targets). Upward pressure on DNP from XNP will cause DNP to rise again. The camera will start refocusing its attention to the forward position in order to decrease XNP until point t=C. At t=C, XNP is has decreased due to the camera's location around the neutral point, and DNP will also start to fall, until the point where XNP begins to rise again the pattern repeats.

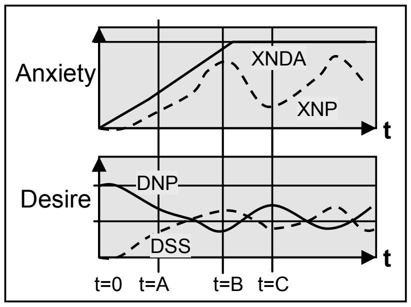


Figure 14. Conceptual Design Point for XNDA, XNP, DNP, DSS

The network diagram for the first iteration is shown in Figure 15, which illustrates the sub-network topology for the ECCS layer.

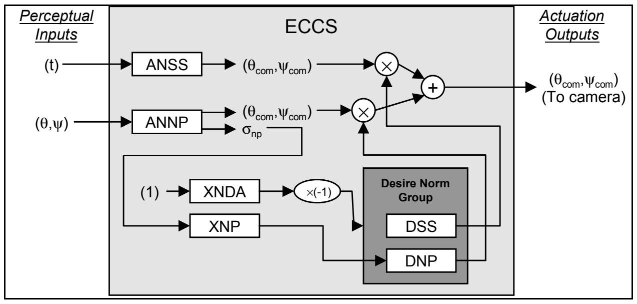


Figure 15. Network Diagram for Iteration 1

The network in Figure 15 was simple enough that the parameters could be tuned by hand and simulated given the scenario described earlier. The resulting system behavior is shown in Figure 16.

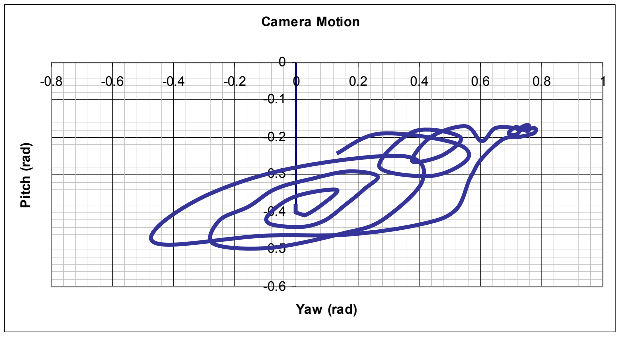


Figure 16 – Simulated results

Iteration 2: Camera Target Following and UAV Command

The second design iteration incorporates a third behavior for the camera: identify anomalies in the video image and command the pan-n-tilt camera system to track the anomalies. A 'detect anomaly' analytical node (ANDA) will process the video input to detect areas of interest, and output camera orientation commands to keep these areas in view. Also, the ANDA will output a normalized parameter for the detection certainty, which measures how prominent the anomaly is on the screen. The emotional network then takes the camera commands from the ANDA into account as the certainty metric increases.

Consider the scenario shown in Figure 17: a camera with fixed position has detected an anomaly ahead of the UAV. The ANDA will begin reporting an increased certainty metric and camera commands to track the anomaly. A desire to track the detected anomaly (DTA) will increase (t=A), and the system will orient the camera to track the anomaly. As the anomaly begins to fall behind the aircraft, the XNP will increase as the camera points further aft. Eventually, at t=B, the XNP will cause the desire DNP to increase beyond the desire to keep tracking the particular anomaly, and the system will start bringing the camera back to the NP. At t=C, the anomaly is no longer detected, and DTA will drop off drastically.

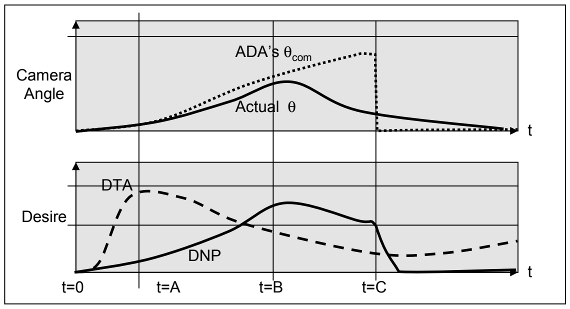


Figure 17. Conceptual Design Point for Anomaly Tracking Behavior

The network design for the second iteration is shown in Figure 18.

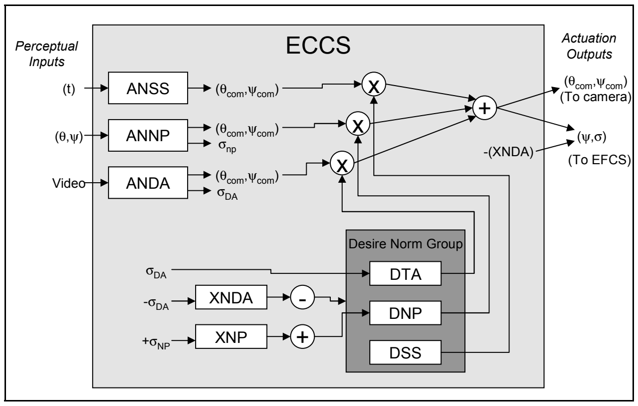


Figure 18. Network Diagram for Iteration 2

System Integration

The ECCS layer was integrated into a complete system with the layers in the composition C_{LB} from Equation (15). The complete system network was instantiated in a simulation of a Mars-class exploration vehicle was created with a pan-n-tilt camera system that renders the scene from the camera's point of view to an off-screen buffer. The camera filters the scene by rendering only one particular texture map, simulating a infrared camera as shown in Figure 19. Here, the bottom-left view port is displaying the contents of the filtered off-screen buffer for debugging. The terrain is textured with a static layer of 'red blotches' that appear in both the rendered display and the off-screen buffer.

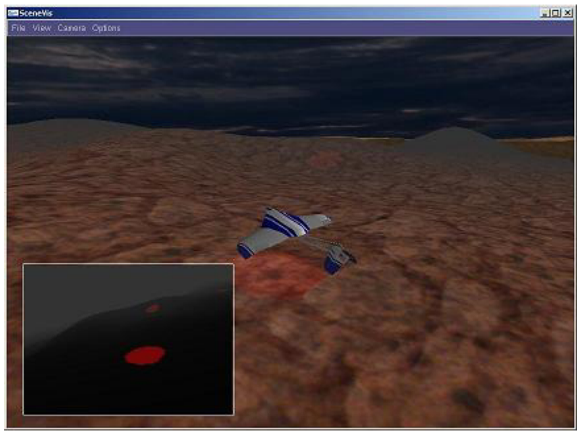


Figure 19 - Pan-N-Tilt Infrared Camera Sensor in Simulation

The simulation was created with multiple configurations of terrain and target locations to select for final parameters tuning. A simulation scenario with resulting trajectory is shown in Figure 20. Two waypoints were created on either side of a hilly area, with a target hidden on the hill sloped so that the target isn't visible from the waypoint path. The aircraft's terrain avoidance anxiety (with a w[e1]:w[e2] ratio of 0.2:0.8) guides the aircraft around the hill till the camera identifies the target. A simple excitation network based on a desire node is used to report the excitation value of the camera to the explorer, and the camera commands blend with the desire for waypoint following behavior and terrain avoidance behaviors. The simulation

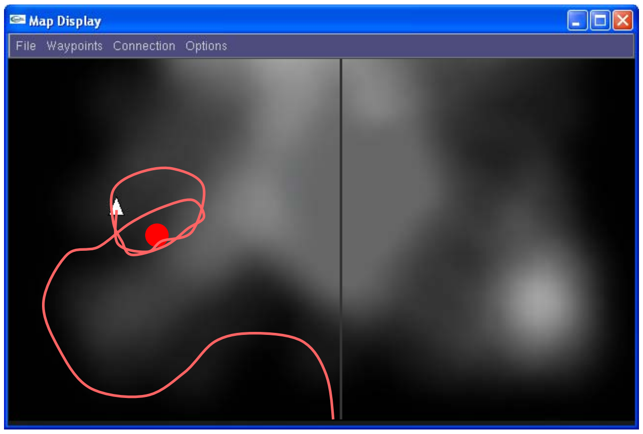


Figure 20 – Simulation Response

6. Memory and Dynamic Networks

The onboard intelligence for the perceptual sub-network of the ECCS described to this point is very reflexive, akin to implementing a complex *vector of desirability*; indeed, the vector of desirability could be defined as the accumulated set of parameter variables of each node in a layer, composition, or system. The camera system to this point responds to current images only by processing based on an impression of the images as they pass in and out of the camera network's short-term memory. Simple desires in a desire group balance the desire to perform a small set of actions in an attempt to maximize the chance of finding useful anomalies, and the size of the anomaly is the only real classification.

The next step taken was to implement a *generalized image model* capable of building a database of impressions of the external world over time. The generalized image model database was implemented in the camera system using dynamic network structures that identify and classify each anomaly encountered. Since anomalies are static, identification was simply a transformation from the camera's screen coordinates to estimate the anomaly's position in the world, which worked well enough for sparse distributions of anomalies (given the complex terrain elevation topology, the transformation method was not precise, and dense clusters of anomalies were often lumped into a single anomaly in the ECCS's memory engram structures).

Given that the ANDA could identify anomalies by locating its real-world position, each new anomaly located was associated with a small emotional memory sub-network that defines several internal state variables dedicated to characterizing the anomaly. We refer to this structure as a **memory engram**. The state variables in an engram at a particular time define the 'impression' that the ECCS has formed about the particular anomaly. These vertices influence the interest metric reported to the EFCS, characterizing how 'excited' the sub-network is about the anomaly. Also, when satisfactorily imaged, any future sightings of the anomaly by the ANDA analytical node are ignored.

This system provides two paths through the emotional system's cognitive structure. A low-level reflexive path provides the system with instinctual motion control of the camera, moving the camera from one anomaly to another based on the systems' higher level state, ignoring certain anomalies and favoring others, while computing low-level camera control commands. Higher-level deliberative paths can also be traced in the network that go through elements of the memory sub-networks to classify the anomalies and compute preferences between different informed and uninformed search strategies, attempting to maximize the utility sensor given limitations on capabilities and little knowledge of the environment.

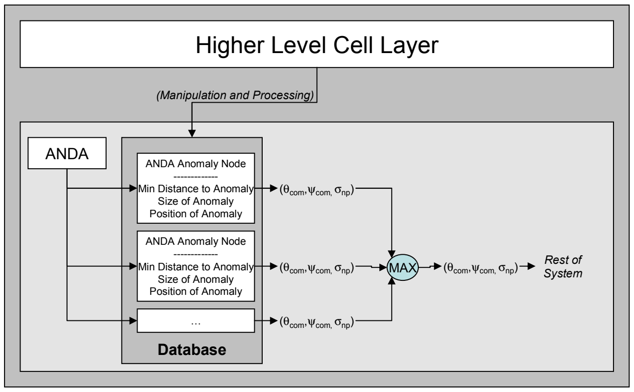


Figure 21. Dynamic Memory Network 'Engrams'

The memory layer modification to the CEL system was added to the complete system and instantiated in simulation in a 'four corners arena', shown in Figure 22. Anomalies were place at four corners of a terrain with a large circular gulley in the middle and two waypoints straddling the circular gulley.

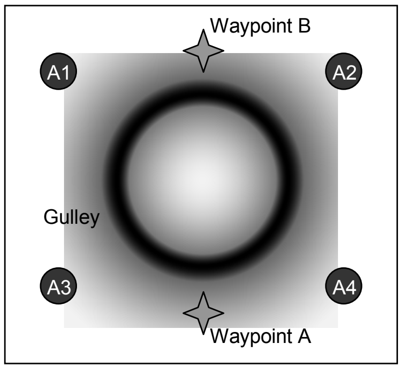


Figure 22 – Four Corners Arena

The results of the simulation are shown in segments in Figure 23 with a w[$\mathbf{e}_1$]:w[$\mathbf{e}_2$] ratio of (0.5:0.5). When a new anomaly is encountered, the camera system becomes excited, guiding the UAV towards the anomaly, until sufficient imaging decreases the excitation level and the UAV continues to the next anomaly. When all anomalies have been sufficiently imaged, the aircraft falls into a stable pattern similar to Figure 9.

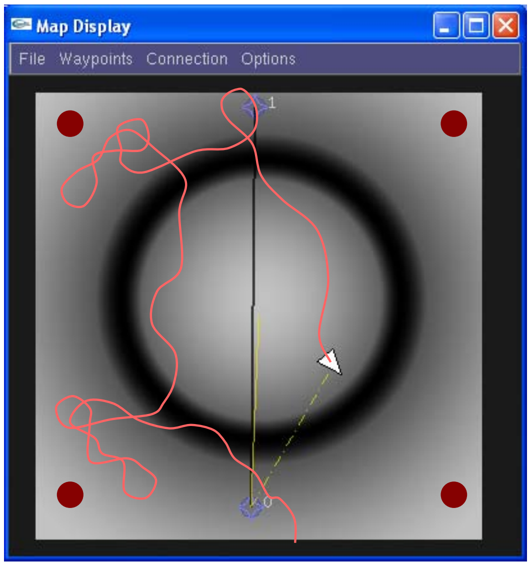


Figure 23 –Trajectory of Simulated Explorer with Engramatic Smart Camera

Other Uses of Dynamic Networks

Given the benefit described that the CEL system has in creating adaptive cognitive networks, several difficulties were encountered. The first is that the cognitive network structures developed are highly dependent on the assumptions. Different assumptions result in different implementations, and changing the assumptions results in a time consuming redesign of individual layers, which in turn requires additional tuning of parameters in different layers when the system is reintegrated. Another related limitation of our current approach is the lack of adaptive learning mechanisms. Behavioral and cognitive networks in this system must be purposefully designed.

Conceivably, learning networks could be purposefully designed to manipulate and reassemble network structures in response to anomalies. Learning 'meta-layers' would adapt the network topology in a predefined manner; these meta-layers could be implemented using the CEL network formulations described, or through a hybridized approach with more traditional techniques.

Dynamic network structures can be defined for an adaptive structure used during the design phase to automatically assemble a design based on techniques such as genetic algorithms. Consider for instance the layer shown in Figure 24. This layer can accommodate any number of ECCS controlled camera systems. This system filters the input from multiple cameras and provides a single output, allowing it to be swapped into the network in place of an existing ECCS system. This intermediary layer provides a deliberative sub-network dedicated to evaluating and calculating preferences for each camera component in a similar manner to how behaviors are combined in the ECCS. This layer also provides a reflexive sub-network that processing and filters the multiple signals into a signal output signal, based on the deliberative layer. Although a simple example, this illustrates how dynamic structures could be used, for instance, to implement simple run-time plug-and-play mechanism, or for use in design time as part of a genetic description of the architecture for manipulation by evolutionary algorithms.

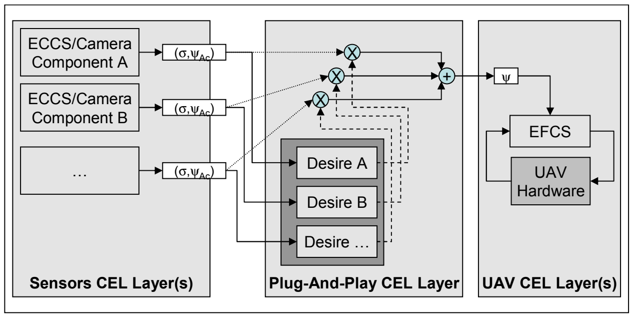


Figure 24. Simple Plug-And-Play Mechanism

7. UAV Navigation System

This section gives a brief description of a more capable lateral mode navigation system created for a remote fixed-wing UAV explorer, part of the autonomous control system being developed. This system was constructed using the iterative approach detailed in previous sections. The major components of this network design are shown in Figure 25. In this design, five different behavioral strategies are available to the explorer: (1) terrain following, (2) random walk, (3) grid search, (4) drift-circling, (5) and camera command. Terrain following and camera command were described previously. Grid searching follows a preplanned grid pattern in the flight management computer's database. Random walk provides random heading commands based on an ad-hoc stochastic algorithm, and the circling command performs a minimal-energy circling drift behavior, intended to minimize control surface actuation and thrust energy expenditure.

A terrain engram layer records properties of the terrain, determining if the ground is appropriate for implementing the terrain following algorithm. Power and fuel levels consumption are monitored, influencing the preference for the minimal-energy circling behavior. The sparseness of targets found negatively influences the desire for behaviors that perform slower searches over smaller areas. Other influences include behavior engrams which record how well-suited a particular behavior is to the environment. Note that the main control input into the UAV is through the ailerons. Control networks for the rudder, throttle, and elevators, while implemented in the CEL system, amount to a typical cascading PID control system, with the main control strategy using elevators to control airspeed and throttle to control altitude AGL.

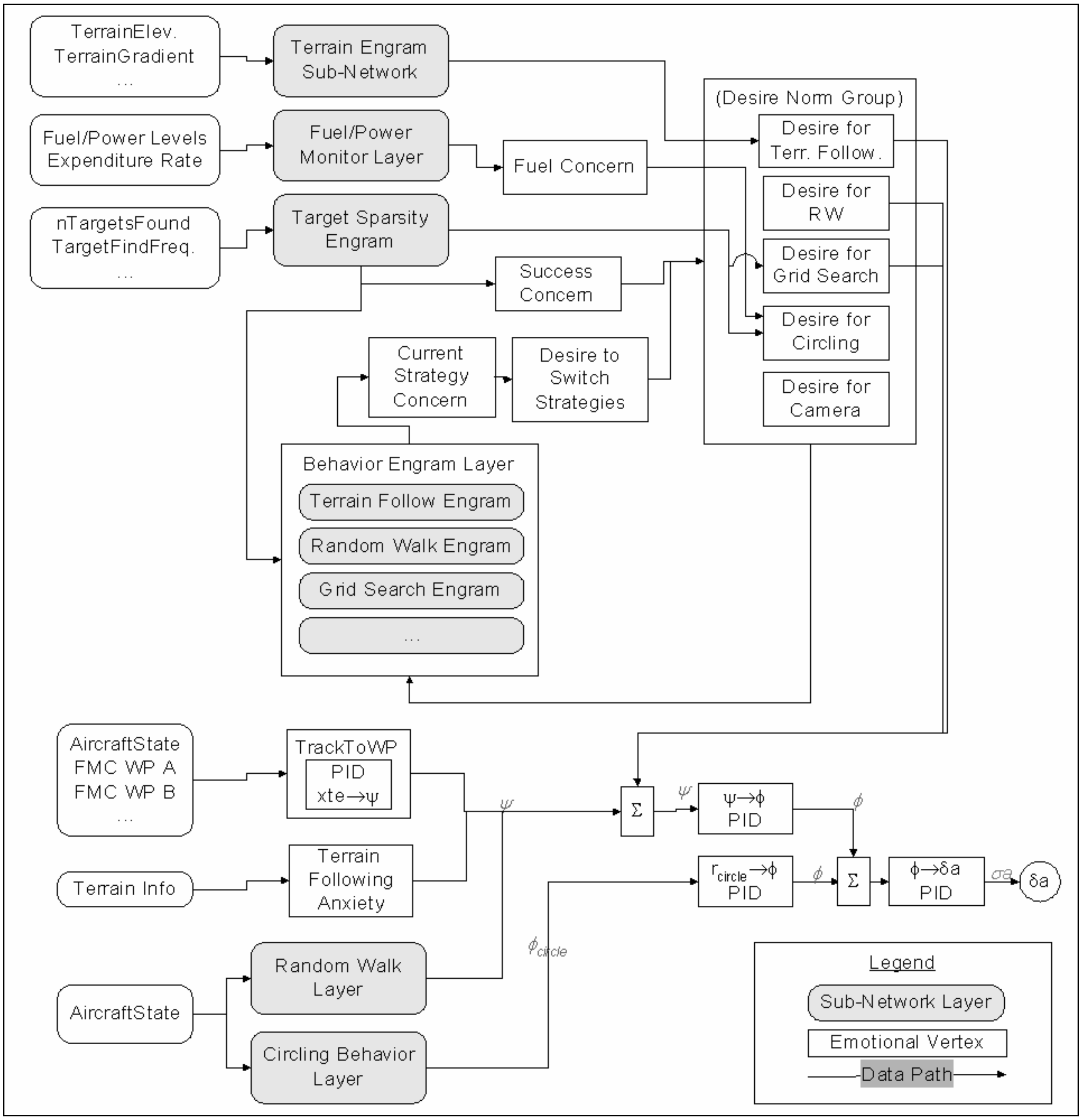


Figure 25 –Lateral UAV Control System

8. CONCLUSION

The Cognitive Emotion Layer architecture provides a structure for implementing emotion based reasoning, adaptive decision making, behavior selection and control for intelligent exploration of remote environments by an autonomous unmanned aerial vehicle. The architecture extends several existing architectures in the literature, expanding them with dynamical formulations that allow systems formulated in the CEL architecture to take on the responsibility of controlling all aspects of behavior, from low-level flight controls to higher level decision-making, adapting its behavior in highly uncertain environments to allow for practical self-governing autonomy. These extensions also provide the ability to analyze the system using system and control theoretic techniques. The CEL formulation accommodates component-based development methodologies for designing and implementing control structures, allowing smaller reusable solutions to be composited quickly into larger networks. Large complex networks can be analyzed and tuned by tracing paths and analyzing influence subgraphs. Further extensions to the dynamic capabilities of the architecture can allow for self-modifying networks that learn and adapt in more efficient manners. However, control systems designed with this system are often rigid and cumbersome to manipulate, requiring substantial redesigns to accommodate small shifts in requirements. This is somewhat alleviated by providing reusable primitive definitions that can be layered and composited, and implementation-specific details can be added to decrease the development time needed to create new systems. The CEL architecture has demonstrated its ability to control small sub-networks and govern simple behaviors in an adaptive manner. Scalability and usability are to be addressed as the system continues to develop.

9. REFERENCES

[1] Clarke, V.C., Jr. "The Ad Hoc Mars Airplane Science Working Group," NASA CR-158000, November 1978.

[2] Girerd, A.R., "The Case for a Robotic Martian Airship," AIAA 97-1460, 1997.

[3] Datta, A, Roget, B., Griffiths, D., Pugliese, G., Sitaraman, J., Bao, J., Liu, L., and Gamard, O., "Design of the Martian Autonomous Rotary-Wing Vehicle," AHS Specialist Meeting on Aerodynamics, Acoustics, and Test and Evaluation, San Francisco, CA, January 2002.

[4] Ippolito, C., Plice, L., Pisanich, G. Holarchical Systems and Emotional Holons: Biologically-Inspired System Designs for Control of Autonomous Aerial Vehicles, *Biologically Inspired Control Systems Conference*, Orlando FL. (July 2003)

[5] Plice, L., Pisanich, G., Young, L., Lau, B. Biologically Inspired 'Behavioral' Strategies for Autonomous Aerial Explorers on Mars, In Proceedings of 2003 IEEE Aero Conference, Big Sky, MT. March 2003

[6] Pisanich, G., Ippolito, C., Plice, L, Young, L., and Lau, B., "Actions, Observations, and Decision-Making: Biologically Inspired Strategies for Autonomous Aerial Vehicles," AIAA Aerospace Sciences Conference, Reno, NV, January 2004.

[7] Pisanich, G., Young, L.A., Ippolito, C., Lau, B., Plice, L., and Lee, P., "Initial Efforts towards Mission-Representative Imaging Surveys from Aerial Explorers," International Society of Optical Engineering (SPIE) Electronic Imaging Conference, San Jose, CA, January 2004.

[8] Young, L.A, Pisanich, G. "Aerial Explorers and Robotic Ecosystems", International Conference on Computing, Communications and Control Technologies: CCCT '04, Austin, Texas, (August 2004).

[9] Damasio, A. R. 1994 Descartes' Error: Emotion, Reason, and the Human Brain. Picador.

[10] LeDoux, J. 1996 *The Emotional Brain*. Simon and Schuster

[11] Picard, R. *Affective Computing*. Random House Publishing.

[12] Ventura, R. and Pinto-Ferreira, C. 1998 *Emotion-Based Agents*. AAAI Publication.

[13] Reilly, S. and Bates, J. *Building Emotional Agents*. Technical Report CMU-CS-92-143. CMU School of Computer Science, Carnegie Mellon University, May 1992

[14] Ortony, A., Clore, G., and Collins, A. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, UK. 1988

[15] Valesquez, J. Modeling emotions and other motivations in synthetic agents. In Proceesings AAAI-97, pages 10-15. AAAI Press and the MIT Press, 1997

[16] Ventura, R., Custodio, L, and Pinto-Ferreira, C. Artificial Emotions, Good Bye Mr. Spock! Instituto de Sistemas e Robotica, Instituto Superior Tecnico, Portugal

[17] McCauley, L and Franklin, S. An Architecture for Emotion. The Institute for Intelligent Systems, University of Memphis.

[18] Selfridge, O. 1959 Pendemonium: A Paradigm for Learning. In Proceedings of the Symposium on Mechanisation of Thought Processs. National Physics Laboratory.

[19] Jackson, J 1987 Idea for a Mind. SIGART Newsletter, no 181 (Jully):23-26

[20] Newell, A. and Simon, H. A. (1976) Computer science as empirical inquire: symbols and search. *Communications of the AMC* 19, 113-126

[21] Beer, R. Dynamical Approaches to Cognitive Science. Trends in Cognitive Science. 2000 Mar; 4(3):91-99

[22] van Gelder, T. J. (1998) "The dynamical hypothesis in cognitive science" in Behavioral and Brain Sciences 21: pp. 615-628.

[23] van Gelder, T. J., & Port, R. (1995) "It's About Time: An Overview of he Dynamical Approach to Cognition." In R. Port & T. van Gelder ed., Mind as Motion: Explorations in the Dynamics of Cognition. Cambridge MA: MIT Press

[24] Bechtel, W. *Dynamics and Decomposition: Are they Compatible?* From the Proceedings of the Australian Cognitive Science Society, 1997

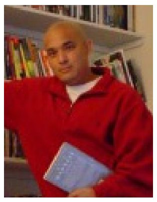
[25] French, R. M & Thomas, E. (1998). The Dynamical Hypothesis: One Battle Behind. *The Behavior and Brain Sciences*, 21: 615-665.

[26] Sciences, 21: 615-665.Casti, L. J. 1992 Reality Rules: Picturing the World in Mathematics (Vols I, II). New York: J. Wiley, 1992

[27] Eliasmith, C. (2001) Attractive and In-Discrete: A critique of two putative virtues of the dynamicist theory of the mind. Minds and Machines. 11: 417-426

[28] Hirsch, M. (1984) The dynamical systems approach to differential equations. Bulletin of American Mathematical Society, **11**, 1-64

10. BIOGRAPHIES



Corey Ippolito *is a Computer Scientist, Aerospace Engineer, and lead developer on the Intelligent Aerial Vehicle project at NASA Ames Research Center. He is a contributing member of AIAA and IEEE, and is affiliated with the NASA Haughton-Mars Project (HMP) and the NASA Biologically-Inspired Engineering for Exploration Systems (BEES) for Mars project. Mr. Ippolito is currently developing small autonomous UAV platforms and control strategies for remote exploration at NASA ARC. He has developed several large-scale architectures, including the Reflection Architecture for embedded system development and simulation, the Cognitive Emotion Layer Architecture for machine intelligence, the Perception Physics Engine for constrained rigid bodies and soft-body physics simulations, the Reconfigurable Flight Simulator, and Self-Assembling Brokering Object Architecture. His background includes control system design, dynamics, simulation, large-scale software architectures, embedded system development, visualization and graphics, and artificial intelligence.*

Greg Pisanich *is a Technical Area Liaison for the QSS Group Inc. within NASA Ames Research Center's Computational Sciences Division and is Project Manager of the Mission Simulation Facility. He holds Master's degrees in Aeronautical Science from Embry Riddle Aeronautical University and Computer Engineering from Santa Clara University. His background and interests include aviation, unmanned aerial vehicles (UAVs), robotics, simulation, autonomy, cognitive modeling, and human factors.*

Larry Young *is a NASA aerospace engineer specializing in advanced concept development for aerial vehicles, autonomous systems, and planetary exploration systems. He is a member of AHS, AIAA and IEEE. He has authored or coauthored over forty publications.*