

# NRASIM: A DESIGN TOOLBOX FOR MULTI-SPACECRAFT INTERFEROMETRIC TELESCOPES

Venkatesh G. Rao and Pierre T. Kabamba  
Department of Aerospace Engineering,  
University of Michigan, Ann Arbor, MI 48109-2140.  
{raov,kabamba}@umich.edu

In this paper we describe the NRASIM toolbox for computer-aided conceptual design of long-baseline interferometric space observatories. The novel features of the design approach implemented by NRASIM are that it permits efficient handling of coupling between structure and behavior in multiagent systems, and permits the use of control-theoretic approaches to mission control design, which speeds up the design process by relieving constraints on constellation architecture.

## Introduction

The objective of this paper is to describe the NRASIM modeling and simulation toolbox for very long baseline space interferometers. The purpose of the toolbox is to provide a conceptual design tool for ongoing research in this area at the University of Michigan\* and the Jet Propulsion Laboratory.

The problem of imaging exosolar planets is currently the focus of significant attention, given NASA's planned series of interferometry missions, culminating in the the Terrestrial Planet Finder (TPF),<sup>1</sup> scheduled for 2012. The focus of this paper is a high-resolution interferometric imaging mission concept with reduced formation-keeping constraints.<sup>2</sup> The concept will require operating several individual space telescopes in dynamically-formed teams with separations on the order of 14,000 km. Current research into this concept is focused on three major, coupled design problems: imaging system design, astrodynamics design and mission control design. The NRASIM toolbox is primarily meant for investigation of the the mission control problem, and the coupling between the astrodynamics and mission control problems.

The novel features of the design approach im-

plemented by NRASIM are that it permits efficient handling of coupling between structure and behavior in multiagent systems (MAS), and permits the use of control-theoretic approaches to mission control design, which speeds up the design process by relieving constraints on structure.

The following sections introduce the problem and our general approach, describe the design space of NRASIM and demonstrate its capabilities with some examples. The last section is an appendix containing some implementation details.

## Problem Formulation

The overall mission design problem may be stated as follows:

*Design a system capable of imaging the surface features of exosolar planets, with a resolution on the order of 1000 km, at distances on the order of 15 parsecs, to maximize the expected science output of the mission, over its lifetime, while staying within budgetary constraints.*

This problem is a much harder version of the problem that is to be solved by missions in the NASA Origins program, such as SIM<sup>3</sup> and TPF.<sup>1</sup> Since the apertures required to image at such high resolutions are of the order of 14000 km across, single aperture systems are not feasible, and sparse

---

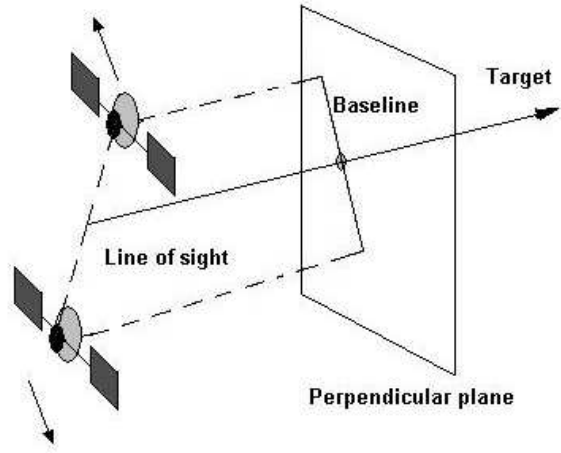
\*This research was funded by NASA grant number NRA-99-05-OSS-0077

aperture systems must be used. This automatically constrains us to an interferometry-based multiagent-systems solution. The particular enabling technology studied in this paper is one based on the exact inverse Huygens Fresnel principle (IHFP)<sup>2</sup> that relieves the extremely tight formation-keeping constraints of other interferometry techniques.

### Sparse Aperture Imaging

We briefly describe the imaging process that is the basis for the class of mission designs that NRASIM has been developed to investigate. At its simplest, an interferometric observation is made by two space telescopes simultaneously measuring the electromagnetic field intensity due to a distant object, at two measurement locations (Figure 1). These measurements can be combined to produce an estimate of the electromagnetic field intensity on a particular patch of a *wavenumber plane*, (a two-dimensional representation of spatial frequency content, similar to the one-dimensional frequency spectrum of time-domain signals). The location of the patch depends on the magnitude and direction of the projection of the relative position vector of the spacecraft onto a plane perpendicular to the line of sight. This projected vector is the *effective baseline*. The actual image can be recovered by making many such measurements at different pairs of locations (thereby ‘covering’ the wavenumber plane), and deconvoluting the resulting wave-number image to provide the physical image (a transformation similar to the process of recovering bitmap representations from JPEG representations).

Imaging exosolar planets requires effective baselines on the order of 14000 km, and additionally requires technology for ‘scrubbing’ the parent starlight. Currently available technologies for this function include local interferometric starlight nulling and coronagraphy. In the architectures being considered in our research, the nulling approach is assumed, wherein a cluster of spacecraft called a ‘nulling pod’, with intra-pod separations on the order of hundreds of meters, provides a scrubbed measurement for interference with another scrubbed measurement made several thousand miles away. Currently NRASIM does not support this level of detail, and only supports individual spacecraft making long-baseline observations in pairs (rather than pairs of clusters).



**Fig. 1 Interferometric Imaging**

The key advantage of IHFP-based imaging is that pairs of observation points can be distributed anywhere within a volume, and are not required to be confined to a plane. This relieves formation keeping constraints significantly.

Since a large number of pairs of observations must be made, and active control of spacecraft position within an observation ‘volume’ is expensive, the current focus of our research group is on designing constellations that achieve the required imaging capabilities through orbital motions alone.

### Multiagent-System Solutions

The problem defined above automatically dictates an MAS solution. MAS solutions can be described in terms of a *structural architecture* and a *behavioral architecture* that, together, describe the static and dynamic properties of the collection of agents that enable it to provide the required functionality. The design of the individual agent (a spacecraft in our case) may reasonably be viewed as a separate, but coupled problem.

We describe some general features of the *process* of finding such solutions, before describing the specific process implemented by NRASIM to find solutions for our application.

Multiagent systems (MAS)<sup>4,5</sup> may be charac-

terized as solutions to engineering design problems that rely on the interactive behavior of multiple autonomous, interacting agents in a physical domain, to achieve the required functionality.

Computing these design solutions from performance requirements is, in general, an analytically intractable problem, mainly because there are multiple representation formalisms involved (continuous, discrete and logical). An additional difficulty in finding MAS design solutions is that prototyping is not an option, except in rare cases,<sup>6</sup> since the system is usually being designed to be built only once, and displays the required functionality only when all the component agents are in place.

Computer modeling and simulation, therefore, are critical to any design process that generates MAS solutions, unlike non-MAS design processes, where computer-aided tools add value, but are often not a necessity.

### Computer-Aided MAS Design

NRASIM implements an instance of a general three-stage computer-aided design process for MAS solutions, which results in the two required design products: a structural architecture and a behavioral architecture. The design process is decomposed into problem solving stages that produce each of these required products, and a preliminary *meta-architecture* stage that constrains the design space by making commitments to particular agent, domain and interaction models.

#### *Design Process Decomposition:*

1. *Meta-Architecture Design:* Meta-architecture design specifies the *types* and *capabilities* of agents to be used as building blocks for the architecture, and identifies the allowable space of deployment configurations. The product of meta-architecture design comprises models of each type of agent, a domain model and an interaction model.
2. *Structural Architecture Design:* Given a meta-architecture, the static structure of the system is specified in terms of a structural architecture by selecting the numbers of the instantiations of agent types and the static features of their

configuration within the domain.

3. *Behavioral Architecture Design:* The static structure of the chosen architecture must be made dynamic by imposing a *behavioral architecture*, which is specified by describing the communication, negotiation, planning, scheduling and control behavior of the system.

For our application, NRASIM supports a particular meta-architecture and a large space of structural and behavioral architecture designs conforming to it.

*Remark 1:* An important feature of this process is progressive constraining. The design process can be viewed as searching the design space spanned by a tree of design choices, making commitments to particular choices for design elements in the order prescribed (in the three stages) during forward phases and backtracking when these choices do not lead to a satisfactory solution. If all choices are found to be unsatisfactory, more options must be added at the topmost level of the tree, expanding the design space. While our ordering of design commitments is not necessary, it follows (for our application) the principle of *least commitment* (fixing the most constrained design element first), which is known to be an efficient approach in any planned process.<sup>7</sup>

*Remark 2:* Note that structure and behavior are coupled. Given a particular performance criterion that is a function of behavior, a behavioral architecture that generates near-optimal behavior with one structure in a class may not do so with a different structure. Conversely, the structure can limit the achievable behavioral performance. In our application, highly redundant structures (symmetric constellations for example) perform better with a given mission control method and certain choices of performance criteria. The observation that an optimization procedure performs better in a subset of the problems to which it is applicable is expected, given emerging results in the ‘no-free-lunch’ optimization literature.<sup>8</sup>

In the next section, we describe the design space that the NRASIM toolbox is designed to help the user search.

## MAS Solutions for Exosolar Imaging

For our application, we define three parameters, *coverage*, *capacity* and *utilization*, that capture the science objectives and budgetary constraints in a general way, which permits the posing of the design problem in the form of a constrained optimization problem, with utility functions and constraints defined in terms of these three parameters.

*Coverage*: is a measure of the range of observations a constellation is capable of making. Coverage is a strong function of types and relative placement of the spacecraft (meta-architecture and structural architecture) and thus encodes both development and production/deployment costs.

*Capacity* is a measure of the throughput of the system, measured, in our formulation, by the *makespan* achieved for the scheduling of representative task sets. Capacity is a strong function of the number of spacecraft (structural architecture) and encodes production and deployment costs

*Utilization* is a measure of the efficiency with which the resources are used. Utilization is a strong function of the mission control design (behavioral architecture), which is a software product specification and encodes software development labor costs, and mission operating costs.

More precise definitions will be presented in the application section. In terms of science objectives, an increase in any of the three parameters will increase the science value of the mission. The design must trade off the three parameters on the tradeoff boundary imposed by external constraints such as budget limits.

Both products of the design process, structural and behavioral architectures, therefore, affect all three parameters encoding design objectives

NRASIM implements certain computer models of one meta-architecture, arbitrary architectures conforming to it, and one class of behavioral architectures, as follows:

**Meta-Architecture Design:** For the VLBI design, meta-architecture problem is the problem of specifying the different spacecraft types (agent mod-

els), the class of orbits and the nature of imaging specifications (domain model), and a characterization of the capabilities of the system, in terms of capacities of various teams, and the constraints on their behavior due to domain features such as eclipsing (interaction model). An implemented system may require two or more types of spacecraft, but currently, NRASIM models only the primary ‘collector’ spacecraft. In terms of orbit classes, ongoing research by other team members<sup>9</sup> has identified circular orbits, containing several spacecraft at different locations, (Fig. 2), as promising, for a variety of reasons. NRASIM currently implements this meta architecture.

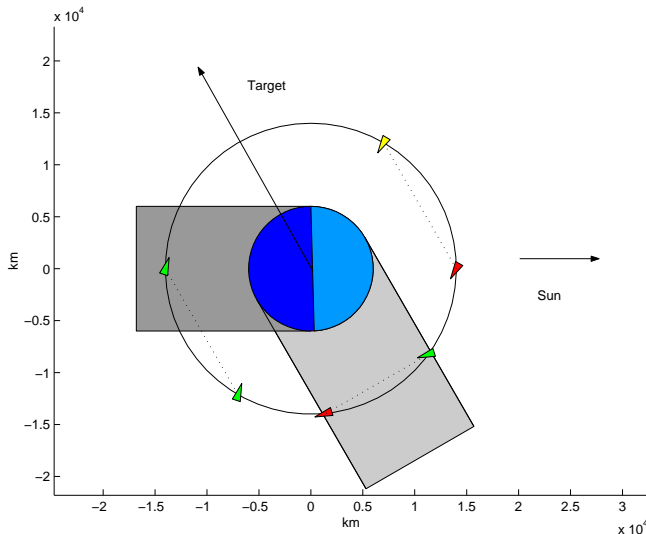
**Structural Architecture Design:** The structural architecture problem can be defined as follows: given a meta-architecture and a behavioral architecture, find a constellation architecture that meets *coverage*, *capacity* and *utilization* constraints, while achieving good performance with respect to a cost function involving these three metrics. NRASIM currently provides restricted support for arbitrary instances of the circular-orbit meta-architecture, the restriction being that only lines of sight in the orbital plane are currently supported.

**Behavioral Architecture:** Given a constellation, find a behavioral architecture that meets *capacity* and *utilization* constraints, while achieving good performance with respect to a cost function involving these two metrics. This problem is, in a sense, a constrained dual of the previous problem, with the structure being fixed and the behavioral architecture being the variable. It is a *constrained* dual because coverage cannot be changed by changing the behavioral architecture alone. NRASIM currently supports the *Mixing Bandit*<sup>10</sup> class of intelligent dispatching algorithms, developed by the authors, which can manage the core behavioral functions of team formation and breakup and observation scheduling.

### Meta-Architecture

In this subsection, we describe the circular-orbits meta-architecture, in terms of three elements: domain, agent and interaction models.

#### *Domain Model*



**Fig. 2 Screenshot from NRASIM**

NRASIM currently implements a simple class of constellation designs whose members comprise  $2m$  identical space telescopes located at different positions in the same circular orbit about a primary<sup>9</sup> (Fig. 2). This class induces a particular format for defining goals for systems in such orbits, as follows.

Recall that an interferometric observation is made by a pair of spacecraft (whose separation is constant in this meta-architecture). A pair can process an observation task only if the projection of the relative position vector of the members of the pair onto a plane perpendicular to the line of sight, (which we have defined as the effective baseline), meets the specifications for that observation. Each imaging task includes specifications on the required baseline, in addition to the obvious specifications of single aperture imaging tasks such as duration, target heading and non-eclipsing.

Currently, the NRASIM toolbox represents the domain goals in terms of a sets of individual imaging tasks for in-plane lines of sight of a particular orbital plane. Each task set  $\mathcal{G}$  comprises  $n$  individual imaging tasks  $\{g_j\}$ , each of which is defined by a quadruple,  $(h_g, d_l, d_u, \tau)$ , comprising the goal heading, lower and upper limits of acceptable baseline lengths, and the imaging duration required. While real-world specifications of imaging tasks are much more complex,<sup>11,12</sup> the purpose of NRASIM is to determine conceptual designs at a high level of abstraction, suitable for refinement. Goal descrip-

tion models used at this stage of design, therefore, are kept as simple as possible.

### Agent Model

Each spacecraft is assumed to have one attitude mode perpendicular to the orbital plane. The model of the  $i^{\text{th}}$  spacecraft is given by

$$\begin{aligned} x_{t+1}^i &= (x_t^i + 1) \bmod N, \\ h_{t+1}^i &= (h_t^i + u_t^i) \bmod N, \\ w_{t+1}^i &= f(w_t^i, u_t^i), \end{aligned} \quad (1)$$

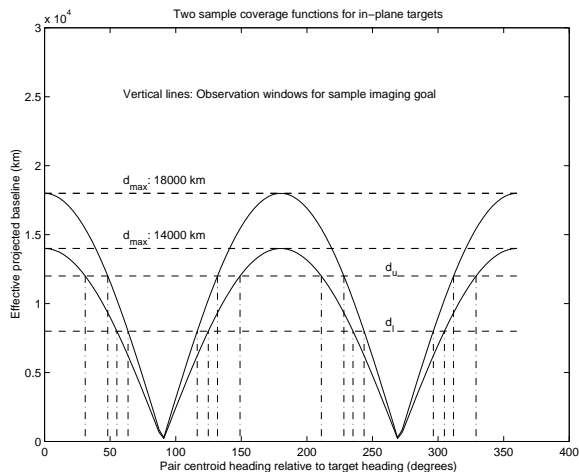
where  $x$  denotes the position of a spacecraft on an  $N$ -step discretization of the circular orbit,  $h$  denotes attitude,  $u$  denotes an attitude control input, and  $w$  the energy of the spacecraft. The  $t$  and  $i$  indices refer, respectively, to time and the identity of the spacecraft. The function  $f$  represents energy dynamics and depends on the spacecraft model. Currently NRASIM supports the full 3-element state space model of each spacecraft, but only uses the first equation (orbital position) in the scheduling. Broadening the interaction of spacecraft models and the scheduling process is part of ongoing development.

### Interaction Model

The domain, as mentioned, comprises a specification of circular orbits and observation task specifications along lines of sight that lie in the orbital plane. An interaction model must specify how agents interact with each other and the domain. The inter-agent interaction is simple, since the relative constellation geometry is fixed, and the capacity of every possible team (2 spacecraft in our meta-architecture) can be described as a function of their relative position vector, which is a periodic function of time. The interaction with the domain consists of eclipsing behavior and additional time constraints imposed by the baseline limits in the goal description.

The baseline limits, eclipsing behavior in circular orbits, relative orbital positions of the spacecraft, and specified duration for a task determine a set of time periods when processing is allowed on a given task by a given pair of spacecraft. These are termed *window constraints*.

These constraints constitute the interaction



**Fig. 3 Coverage functions for two different pairs. Note that the pair with lower maximum separation has longer windows for the same range of baselines. Eclipses are not shown.**

model, since they specify if and when a particular pair of spacecraft may be used for a particular task. The constraints can be represented in the form of a set of *coverage functions* that map to all possible pairings of spacecraft.

Let the coverage function of a pair of spacecraft  $(a, b)$  be defined by the instantaneous effective baseline generated by the pair at a particular position, for a particular target heading (Fig. 3 shows the coverage of two different pairs for one orbital period, as a function of the heading of the centroid of the pair). For our meta-architecture, the coverage function is simply the absolute value of a sinusoid, with a maximum of  $d_{\max}(a, b)$ , the separation of the two spacecraft. Given a particular goal, the coverage function can be used to compute the locations of observation windows.

Note that the difference in the capacities of two pairs depends on the difference between their maximum baselines. Closely-spaced pairs allow longer observations for shorter ranges of baselines, producing a ‘breadth-versus-depth’ tradeoff in coverage capabilities of different pairs.

Given this meta architecture, *the structural architecture design problem involves selecting the circular orbit and the number and placement of spacecraft in it.*

*The behavioral architecture problem involves*

*finding an algorithm that assigns tasks and forms and breaks up teams.*

## Applications

In this section we describe how the two products of the process: structural and behavioral architectures, are determined, given the meta-architecture. The process is to simply fix one of the two and search for candidates that satisfy the requirements in the space of the free design variable. We can therefore fix structure and seek optimal behavior, or fix behavior and seek optimal structure. NRASIM supports this design process by providing separate modules for specifying constellations, goal sets and schedulers (behavioral architectures).

### Optimizing Structure

In this section, we demonstrate, by means of an example, the use of NRASIM in determining optimal selection of circular orbits and placement of spacecraft within them. A detailed development can be found in a companion article.<sup>13</sup>

We defined the problem of structural architecture design as the problem of specifying a circular orbit and the relative positions of the spacecraft within it, to satisfy coverage, capacity and utilization requirements. We refine the definitions of these parameters, with reference to our meta-architecture, as follows:

*Coverage:* is specified in terms of the range of observations that a particular canonical constellation,  $s^*$ , is capable of making. This constellation is automatically a candidate solution, and all candidate solutions must be capable of equal or greater coverage.

*Capacity:* is specified in terms of the average, (or best or worst) expected schedule length,  $L$  (or *makespan*) achieved by the constellation, with a particular scheduler, for task sets of size  $n$  in the coverage range.

*Utilization:* is specified in terms of a lower bound on a quality parameter  $q$ , defined as  $2 \sum p_i / mL$ , where  $p_i$  are the individual processing times of the  $n$  jobs,  $m$  is the number of agents and  $L$  is the schedule

length. This quantity is the ratio of a lower bound on the schedule length (which would be achieved by a zero-slack perfectly load balanced schedule if one were possible), to the actual schedule length. The range of  $q$  is the interval  $(0, 1]$ .

We therefore state the structural architecture problem as follows.

*Given performance requirements in the form of a canonical constellation the coverage of  $s^*$  and quality and makespan bounds  $q^*$  and  $L^*$  (for  $n$  jobs), and a scheduler  $M$ , find a constellation which guarantees satisfaction of the performance requirements.*

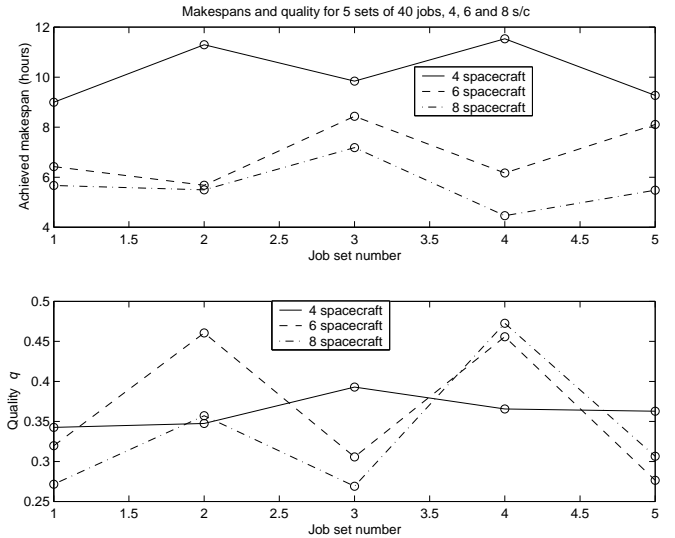
The problem is analytically intractable, since absolute performance guarantees are hard to compute in most scheduling problems. NRASIM facilitates the computation of solutions which achieve the required performance in a probabilistic sense, i.e., we can determine whether or not a candidate constellation performs as required based on scheduling results on a set of sample task sets. By interpreting the bounds as requirements on the best, worst or average performance over this finite set of sample runs, we can solve the problem heuristically. More systematic Monte Carlo experiments can also be conducted using NRASIM.

*Example:* Consider a canonical constellation  $s^*$ , comprising four spacecraft in an 18000 km earth orbit, evenly spaced 10 degrees apart. Consider 3 candidate constellations that have equal or greater coverage than  $s^*$ , all of which are in the same 18000 km orbit. The first,  $s_1$ , is identical with  $s^*$ . The second,  $s_2$ , consists of six spacecraft with the same 10 degree separations, and the third,  $s_3$ , consists of 8 spacecraft with 10 degree separations.

Figure 4 shows the results of applying a simple, greedy scheduler from the Mixing-Bandits class,<sup>10</sup> for five task sets of size 40 each, and Table 1 lists the constellations that satisfy various requirements  $L^*$  and  $q^*$ , based on these results, in a worst-case sense. Note the tradeoff between capacity and utilization, as the former is increased by increasing capacity.

### Optimizing Behavior

The behavioral architecture design problem can be stated as follows:



**Fig. 4 Structural Architecture Design**

Objectives	Acceptable Solutions
1 $L^*=12$ hours, $q^*=0.34$	$s_1$
2 $L^*=8$ hours, $q^*=0.4$	None
3 $L^*=9$ hours, $q^*=0.25$	$s_2, s_3$
4 $L^*=12$ hours, $q^*=0.25$	All three

**Table 1 Structural architecture solutions**

*Given a constellation  $s$  and performance requirements in the form of makespan and quality bounds  $L^*$  (for  $n$  tasks) and  $q^*$ , find a scheduling method  $M$  that guarantees satisfaction of the performance objectives.*

As before, this problem is analytically intractable, and we seek probabilistic satisfaction of the objectives using sample tests.

NRASIM currently supports one class of *Mixing-Bandit*<sup>10</sup> intelligent dispatching algorithms for team management and observation scheduling. These algorithms create the schedule by adding one task at a time, in strict temporal order. The process comprises three levels of decision-making, which mimic bandit processes.<sup>14</sup>

We summarize the results of a design experiment for a constellation comprising 4 spacecraft evenly distributed over a 90 degree orbital arc, in an 18000 km earth orbit. The experiment compared four schedulers of increasing complexity, for a representative task set of 300 tasks, generated in the coverage range as a uniform random distribution.

Details are under review for publication.<sup>10</sup> The candidate methods compared were as follows.

1. *Greedy*: This method uses a greedy (minimum slack) dispatching, with random control of team formation and breakup.
2. *Two-Phase*: This method uses reinforcement learning to identify the effectiveness of various team configurations during an exploration phase comprising the first  $k$  percent of assignments, and preferentially creates these configurations during an exploitation phase.
3. *Two-Phase with rapid exploration*: this method extends the previous method by forcing rapid changes in the team configurations during exploration, to gather a larger amount of effectiveness data.
4. *Adaptive*: This method uses a continuous learning process with discounted past, to track changes in the statistical distribution of remaining unscheduled goals instead of a fixed demarcation of exploration and exploitation phases.

Table 2 shows the best-of-three-attempts results for the four methods, computed using NRASIM. Overall, the most sophisticated scheduler reduced makespan by 21% relative to the least sophisticated controller. (explicit values of  $L^*$  and  $q^*$  were not prescribed beforehand, since this test was aimed at discovering the capabilities of the schedulers).

Method	Best Makespan (hours)	Best $q$	% change (makespan)
1.	54.41	0.592	0%
2.	48.42	0.665	-11%
3.	47.16	0.683	-13.3%
4.	42.67	0.755	-21.6%

**Table 2 Comparison of methods**

*Remark 3:* From the structural and behavioral architecture design examples, it should be clear how NRASIM facilitates flexible iterative solution of these two problems by allowing the user to fix either structure or behavior in a set of simulation experiments.

*Remark 4:* The example presented used one common control technique, adaptive control, to drive the

scheduling. Since this process ‘learns’ the capabilities of any architecture, using this method permits greater flexibility in choosing a constellation. This results in fewer iterations being required to fine-tune a design. The adaptive scheme used does not explicitly use feedback from the remaining, unscheduled tasks to control the scheduling, but NRASIM and the Mixing-Bandits class of algorithms both provide support for such feedback-driven scheduling as well, and this is the subject of ongoing work.

## Conclusions

In this paper, we formulated the high-resolution exosolar planet imaging problem and showed how a general description of MAS solutions, in terms of structural and behavioral architectures, leads to a three-stage design process for finding the right combination of these two elements to satisfy design requirements. The capabilities of the NRASIM toolbox to enable a computer-aided implementation of this design process, for our application, were demonstrated, with reference to specific examples.

Two novel features of NRASIM were demonstrated. First we demonstrated a capacity for accommodating strong coupling between the structural and behavioral design problems, by permitting flexible iteration between the two dual problems. Second, the toolbox provides functionality for exploring control-theoretic approaches to designing the behavioral architecture, which, due to their greater capacity for handling uncertainty compared to classic computer science approaches, speeds up the design process, by providing greater freedom for structural architecture design.

Future work will include extension of the capabilities of NRASIM to handle more powerful meta-architectures, covering more complex agent models and different orbit designs.

## Appendix: Implementation Details

NRASIM has been implemented as a library of MATLAB routines. The core functionality can be described with reference to the activity diagram in Fig. 5. Constellations are created and saved as configuration data files by following the path 1 – 2 – 4 – 5 – 7. Task sets are created with reference to a



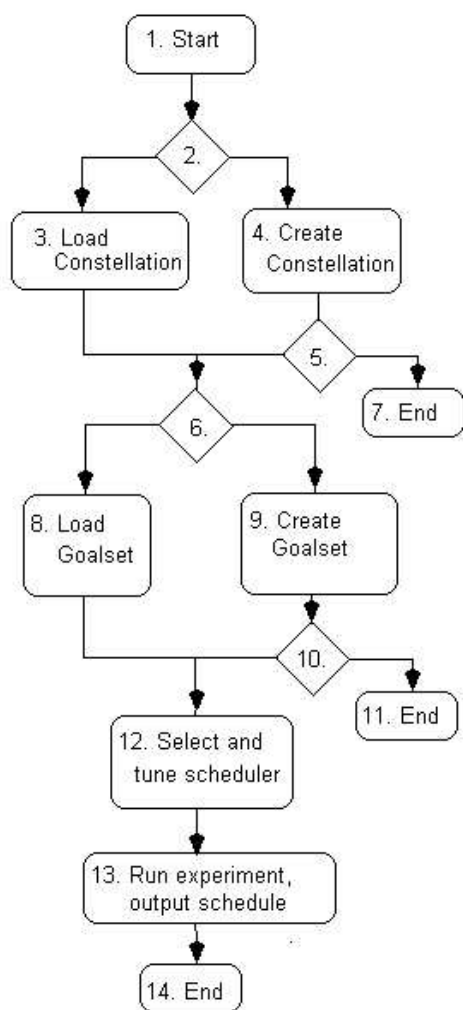


Fig. 5 Design Flowchart

particular constellation, following the sequence 1 – 2 – 3 – 6 – 9 – 10 – 11. Experiments are run following sequence 1 – 2 – 3 – 6 – 8 – 12 – 13 – 14. Alternate paths through the Fig. 5 can create configurations and data sets on the fly for one-off experiments.

The functionality of Fig. 5 is achieved by modular units. The functionality in blocks 1-11 is straightforward, and is provided by a set of architecture modules that produce an appropriate discrete description of agents, domain and interaction (coverage functions) from the user-provided constellation parameters (radius and orbital positions). Blocks 12 and 13 implement the core logic of the scheduling process. Currently, NRASIM implements a general-purpose constraint posting module, which computes

window constraints to generate feasible assignments for any time for which it is invoked. This module, combined with one of a library of intelligent dispatchers,<sup>10</sup> provides the scheduling capability. The constraint module can handle non-dispatching scheduling methods as well.

The schedule and assignment data, along with constellation geometry, and a spacecraft state space model, are sufficient to generate state trajectories of the system. NRASIM provides tools to generate state trajectories from a computed schedule, and a visualization module to view the resultant system behavior through animations (Fig. 2 is a screenshot). Selected schedule analysis tools, specific to the problems being investigated, have also been implemented to detect, for instance, any preferential assignment behavior generated by a given scheduler. Currently, the implementation does not support iterative repair/rescheduling schemes or arbitrary constraints, and providing capacity for this functionality is the focus of ongoing work. This is being done by rewriting the constraints module as a generalized blackboard mechanism that permit arbitrary modules, such as a concurrent spacecraft energy system simulation, to constraint ongoing scheduling.

The current implementation of NRASIM is mainly intended to facilitate design for quantitative performance objectives, and MATLAB was found to be the best platform, since it has a rich set of numerical and visualization tools to support quantitative design processes, in particular control-theoretic design. The disadvantage of a MATLAB implementation is that it provides limited capacity for scaling, due to the fundamentally single-threaded and procedure-oriented nature of the MATLAB platform. Extending NRASIM to handle such issues as inter-agent communication and messaging, unanticipated interrupts and more general asynchronous process models will require an object-oriented implementation.

## References

- <sup>1</sup>Beichman, C. A., Woolf, N. J., and (Eds.), C. A. L., "The Terrestrial Planet Finder: A Nasa Origins Program to Search for Habitable Planets," *JPL-Publication*, Vol. 99, No. 3, May 1999.
- <sup>2</sup>Hyland, D. C., "Interferometric Imaging Concepts with Reduced Formation-Keeping Constraints," *AIAA Space 2001 Conference*, Albuquerque, NM, August 2001.

<sup>3</sup>Unwin, S. C. and Shao, M., "Space Interferometry Mission," *Proceedings of SPIE*, Munich, Germany, 2000, pp. 754–761.

<sup>4</sup>Weiss, G., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge, MA, 2000.

<sup>5</sup>Wooldridge, M., *Introduction to Multiagent Systems*, John Wiley, 2002.

<sup>6</sup>Knight, W., "Military robots to get swarm intelligence," *New Scientist*, April 2003.

<sup>7</sup>Weld, D. S., "Recent Advances in AI Planning," *AI Magazine*, Vol. 20, No. 2, 1999, pp. 93–123.

<sup>8</sup>Wolpert, D. H. and Macready, W. G., "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computing*, Vol. 1, No. 1, April 1997, pp. 67–82.

<sup>9</sup>Hussain, I. I., Scheeres, D. J., and Hyland, D. C., "Interferometric Observatories in Earth Orbit," *Proc. AAS/AIAA Space Flight Mechanics Meeting*, Ponce, PR, February 2003.

<sup>10</sup>Rao, V. G. and Kabamba, P. T., "Mixing-Bandit Algorithms for Multiagent Team Management and Scheduling: Application to Interferometric Observatories," *2003 IEEE Conf. on Decision and Control (submitted)*.

<sup>11</sup>Muscettola, N., "HSTS: Integrating Planning and Scheduling," *Intelligent Scheduling*, edited by M. Zweben and M. Fox, Morgan Kaufmann, 1994.

<sup>12</sup>Johnston, M. D. and Miller, G. E., "Intelligent Scheduling of Hubble Space Telescope Observations," *Intelligent Scheduling*, edited by M. Zweben and M. S. Fox.

<sup>13</sup>Rao, V. G. and Kabamba, P. T., "Interferometric Observatories in Circular Orbits: Designing Constellations for Capacity, Coverage and Utilization," *2003 AAS/AIAA Astrodynamics Specialists Conference*, Big Sky, Montana, August 2003.

<sup>14</sup>Berry, D. A. and Fristedt, B., *Bandit Problems: Sequential Allocation of Experiments*, Chapman and Hall, 1985.