Co-Generation of Design and Control for Dynamical Systems

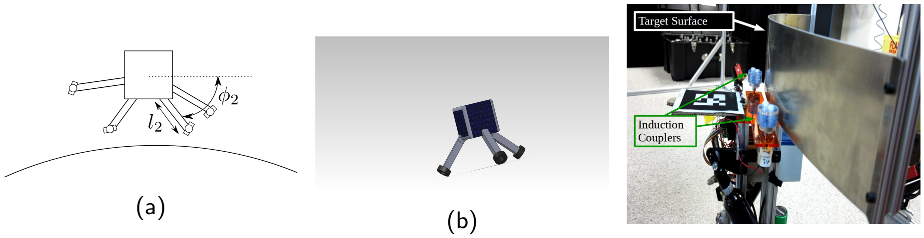
Benjamin Reinhardt, Ross Knepper, Mason Peck

Abstract

This paper describes a method for generating the parameterized design of a complex dynamical system based on a co-generated controller. The algorithm generates and evaluates a design based on the volume of state space that can be stabilized by its associated controller. This volume acts as a conservative metric for the extensibility of the generated controller and by proxy, the quality of the design. The approach is motivated specifically by a robotic inspection spacecraft using novel actuators. It exhibits complex behaviors that couple its design and possible controllers. This approach addresses the weaknesses in traditional techniques when designing high-dimensional dynamical systems that have actuator saturation and uncontrollable swaths of state-space coupled with the system's design parameters. The algorithm takes a mutable, parameterized design and a controller prototype as input. It probabilistically searches for the design-controller combination with the best control volume, then outputs a design and controller. With minimal human intervention, this approach can generate high-dimensional designs with several parameters that are unsuitable for other design and controller synthesis methods. We demonstrate the system's efficacy through two simulated systems - a double pendulum and an inspection spacecraft.

1 Introduction

The goal of this paper is to introduce a new approach to algorithmic dynamical system design. Our probabilistic co-generation method uses techniques from evolutionary design, motion planning, and controller verification to generate designs and evaluate them on controller coverage. This process results in physical design parameters and a multi-query controller. These together attempt to maximize the volume of state-space that the system can explore without becoming unstable. Algorithmically generated designs that can safely reach many goal states are especially valuable to our motivating system: an inspector spacecraft using a novel actuation technology Reinhardt & Peck (2015). The inspector exhibits a confluence of problematic characteristics for traditional co-generation methods - complex dynamics, multiple goal states, and high dimensionality. While designed for a specific system, this method is broadly applicable to the design of systems with similar characteristics.



(a) (b) (c)

Fig. 1: Three views of an electromagnetically actuated inspection spacecraft that is an ideal target for probabilistic co-generation. Left to right - (1a) Schematic diagram showing a small set of possible system parameters. (1b) Conceptual image. (1c) Prototype system showing the conductive surface and actuators.

1.1 Simultaneous Design and Control

Many systems are full of numerical design parameters that dominate their dynamics. These parameters can be anything from the wing width of an airplane to the volume of a chemical tank to the length of a pendulum or the number of joints in an arm. The resulting dynamic effects are then unavoidably linked to the system's control law. A naive set of parameters can render the system effectively useless. Ideally, the system's parameters would take its future controller into account to maximize its whole performance.

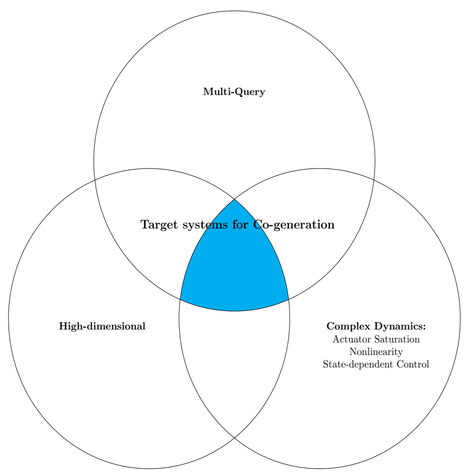


Fig. 2: Probabilistic co-generation addresses high-dimensional, multi-query systems with complex dynamics.

System design can be long and involved, so ideally it would leverage increased computing power to expedite the process and unlock the power of complex systems. Traditional methods for designing simple systems close the loop between the parameters and controller through a combination of intuition, analysis/tuning, and leveraging well-studied systems. This process tends to fall apart as the system becomes more novel or complex - a perfect place for more algorithmic approaches to step in Yao & Higuchi (1999). As additive manufacturing, laser cutters, and cheap, multi-use electronics make new and on-the-fly system creation more common, design techniques need to match these advances Gowdy & a.a. Rizzi (1999).

Note that many systems don't need algorithmic design synthesis. Often, the design of a physical system is determined by external factors - manufacturing costs and well-understood designs are important considerations. A novel system is often intuitive or well understood; it may have a few free parameters which have well-known effects on performance Hurst et al. (2007). However, even in these cases algorithmic system generation enables a symbiotic pairing between humans and machines - leaving the humans to do the creative work they are best at, while the computer takes care of the details.

1.2 Approaches

Different fields have used algorithmic co-generation to design parameters and controllers across several applications. Chemical engineering uses several tools to simultaneously generate designs and control laws for large-scale systems Kookos & Perkins (2001); Ricardez-Sandoval et al. (2010, 2009); Bansal et al. (2000). Aerospace engineers have applied co-generation to the designs of flexible spacecraft Hale et al. (1985); Ravichandran et al. (2006); Sobieszczanski-Sobieski & Haftka (1997) and micro air vehicles before they were drones Kajiwara, I.; Haftka (1999). The co-generation methods used in aerospace and chemical engineering take a purely optimization-based approach, which can provide explicit performance guarantees. These guarantees are needed because the systems in question are often delicate and expensive.

Algorithmic co-generation can be especially useful in robotics, where systems need to adapt to many situations Ravichandran et al. (2006) and may be reconfigurable Kotay et al. (1998) or designed on the fly Yim et al. (2007). Robotic co-generation uses either optimization over the parameterized dynamics or genetic algorithms paired with simulations Auerbach & Bongard (2010); Hornby et al. (2003).

Each of these systems has trade-offs. Optimization-based methods perform well in several cases: when the dynamics can be expressed analytically Rastegar et al. (1999) or when the dynamics are smooth, low dimensional or linear. If a system doesn't meet these conditions, the optimization could fail to converge or require excessive tuning (essentially reverting to hand-design.) Simulation-based co-generation can handle systems that defy traditional optimization. However, whereas simulations can demonstrate functionality, they give no guarantees. Additionally, both optimization and simulation-based methods primarily target systems with a single goal - whether it be a point in state space or a limit cycle. Stochastic MPC can provide guarantees and deal with nonlinearities, but still depends on a single query system and converging dynamics Bahakim & Ricardez-Sandoval (2014).

Why is yet another approach needed? New technologies enable systems that fall through the holes in previous methods. For example, the robotic inspection spacecraft that motivates the probabilistic co-generation approach Reinhardt & Peck (2014). Unlike a free-flying inspection vehicle Choset et al. (1999), it uses novel electromagnetic actuators that allow it to grapple surfaces without mechanical contact. These actuators have advantages, requiring no propellant, but a poorly designed controller will send the inspector spinning into space. The probabilistic co-generation approach can unlock multi-query controllers for an entire class of systems with nonlinear, high-dimensional dynamics and limited control authority (figure 2.)

1.3 Building Blocks

Complex systems need tools to tackle several separate problems: generating designs and multi-controllers for high dimensional and complex dynamics; providing some guarantees on performance; and evaluating the whole combination. To do this, the probabilistic co-generation approach draws on work in several areas: probabilistic path planning, algorithmic control verification, and evolutionary design.

Probabilistic methods like Rapidly-exploring Random Trees (RRTs) LaValle & Kuffner, J.J. (2001) are a common approach for attacking high dimensional systems. Several methods have demonstrated how RRTs can control systems with non-linear dynamics Perez et al. (2012) and non-holonomicKaraman & Frazzoli (2013). Probabilistic methods combine naturally with planning approaches that use local controllers to cover state space. These include sequential potential functions for navigation Conner et al. (2003) and preimage backchaining, which uses sequential control 'funnels' to drive a system to a goal state Lozano-Perez et al. (1984); Mason (1985); Burridge et al. (1999).

The controller half of probabilistic co-generation is heavily influenced by LQR trees Tedrake et al. (2010), which combine sequential stabilizing funnels with probabilistic methods and numerical verification Tobenkin et al. (2011). Additionally, LQR trees provide coverage over all (reachable) state space by finding the regions of attraction for their control gains through SoS verification Reist & Tedrake (2010). Our approach uses similarly verified coverage as information about the state space volume that a controller can drive to a stable goal state. Like LQR trees, our approach uses Linear Quadratic Regulator (LQR) control gains Kwakernaak & Sivan (1972) in each individual controller. While simple, LQR gains can be quickly calculated and extended for control-limited systems Scokaert & Rawlings (1998).

RRTs in general and LQR trees in particular are single-query approaches - they drive the system to a single goal state. Single-query planners are good for a single task or maneuver, but a system that may need to perform several different tasks needs a multi-query planner. There remain fewer analogs in dynamical systems for multi-query probabilistic planners like Probabilistic Road Maps (PRMs) Kavraki et al. (1996); Mitchell et al. (2005) than their single-query counterparts because many dynamical systems have only a few stable goal states. Thus, verification and controller generation for multi-query dynamical systems is an area of active research Majumdar (2012). Following the lead of PRMs, our approach creates a safe roadmap through state space by connecting stable goal states with overlapping Region of Attractions (RoAs).

The plethora of possible goal states raise the question of how to evaluate the controller, because it can't be judged on driving the system to any single state.

The number of states it can stabilize provides a different metric. Ideally, this metric would be the volume of the closed loop system's reachable set. Many systems can use the solution to the Hamilton-Jacobi-Bellman (HJB) equation to find their reachable set Mitchell et al. (2005). Other methods of verification include 'growing' reachable cells Conner et al. (2006). However, these methods are computationally intractable above three or four dimensions which makes them impractical even for dynamical systems that are unconstrained in three degrees of freedom.

Instead, our system needs an approximation for the full reachable set. One possible approximation is the volume covered by the RoAsof each set of control gains. Input-constrained systems without a clear metric for total energy defy traditional energy-based Lyapunov functions Koditschek (1989), so our method needs to generate Lyapunov functions by another path. Sum-of-Squares (SOS) programming leverages numerical optimization and increased computational power to algorithmically find a Lyapunov function for a system Papachristodoulou, Antonis; Prajna (2002). The function's level sets define a RoA in state space where the controller can always stabilize the system. These RoA are often small in strongly nonlinear systems because a single LQR controller is based on a linearization of the system. One downside of SOS programming is that it is computationally intensive and scales poorly to high dimensional systems. However, recent research has shown that Diagonally-dominant Sum-of-Squares (DSOS) programming and Scaled-Diagonally-dominant Sum-of-Squares (SDSOS) programming are viable alternatives to SOS programming in high dimensions Ahmadi & Parrilo (2014).

In short, complicated systems require creative solutions. Our approach draws from research in probabilistic path planning, numerical verification, and simultaneous design and control in several fields. Motivated by a robotic inspection spacecraft, it simultaneously synthesize designs and controllers for complex systems.

2 System Description

This probabilistic co-generation method is meant for complex dynamical systems where other methods fall short. In general, these systems have nonlinear, state-dependent actuators, a high dimensional state space, and many stable points. Consider a many-link pendulum with limited actuators at each joint. The actuators are sufficient to hold the joints in many stable configurations, but other states can't be kept from chaotic motion. More complicated real-world examples include some underactuated walkers Hurst et al. (2007) and many electromagnetically actuated systems Shoer & Peck (2013) including the inspection spacecraft described in subsection 2.2.

2.1 General System Description

Co-generation targets systems with parameterized designs, $\mathcal{D}(\mathbf{p})$ where $\mathbf{p}$ are mutable and numerical parameters of the system. In the thought experiment,

p are the length and mass of each pendulum link. These parameters are either discrete (number of links) or continuous (properties of the links.) $\mathcal{D}$, along with fixed parameters and information about the environment, must have enough information to numerically model the system.

$\mathcal{D}$ and the independent environment variables[1] specify the nonlinear dynamics of the system with a state $\mathbf{x}$ and control input $\mathbf{u}$.

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{1}$$

A controller $\mathcal{C}$ defines a set of control actions $\mathbf{u}$(t,$\mathbf{x}$) that will drive the system to a stable goal state, $\mathbf{x}$_G. The total volume of the regions of attraction in $\mathcal{C}$ define V($\mathcal{C}$) - the volume of state space that $\mathcal{C}$ can successfully drive to some $\mathbf{x}$_G. While V($\mathcal{C}$) has little meaning in an absolute sense, it provides a metric to compare different $\mathcal{D}$ based on how much state space they can explore before "wandering" into a possibly unrecoverable state.

In general, the co-generation problem is to find a $\mathcal{D}$ and $\mathcal{C}$ that create a system that exhibits some desired performance. In the toy example, performance is based on possible joint configurations and in the motivating example, performance is based on controllable rigid-body configurations. The larger the set of achievable configurations, the better the performance.

Traditional human-intuited or optimization-based design and controllers fall short when there exists a combination of:

- $\mathbf{x}$ is high dimensional.
- $\mathbf{p}$ is high dimensional.
- f($\mathbf{x}$,$\mathbf{u}$) is nonlinear and not analytically solvable.
- $\mathbf{u}$ saturates, especially if $\mathbf{u}_{max}$ and $\mathbf{u}_{min}$ are functions of $\mathbf{x}$.
- The system cannot be controlled over all of state space, regardless of the design.
- Unknown goal specifications and a large number of possible goal points.

The dimensionality of **p** and **x** that constitute 'high dimensional' varies based on the behavior of the other criteria. High dimensional systems with slowly changing dynamics and state-independent actuators are often still amenable to optimization-based design. At some dimension, human intuition breaks down and optimizers fail to converge. In these cases, a probabilistic approach to both the design and control may be needed.

[1] Independent environment variables can be fixed design parameters (like a fixed mass) or external factors that affect the dynamics, like an external magnetic field

2.2 Motivating System Description

The real-world system motivating this approach is a robotic inspection spacecraft using induction couplers for contactless manipulation. Induction couplers create actuation forces from the interaction between time-varying magnetic fields of spinning magnets or oscillating electromagnets and induced currents in conductive targets. These forces allow a spacecraft to manipulate a target and locomote itself without physical contact at distances of several centimeters.

Induction couplers enable unique capabilities, but also exhibit the behaviors that make traditional design-and-then-control difficult. The rigid body system itself has a 12 dimensional state space. Each actuator could be placed anywhere on the inspector, with any orientation - six parameters per actuator. The actuation forces both increase nonlinearly and change direction with the control input. The actuation force's direction and magnitude also depend on the distance and relative orientation between the surface and the actuator. With enough distance, the actuators become effectively useless which can cause the system to become uncontrollable if too many actuators are too far from the target. Finally, a robotic inspector needs the ability to stabilize itself in many different configurations. With this complexity and number of free parameters, creating a design without considering its associated controller could lead to systems with poor performance. The actuators' nonlinear, state-dependent, and saturated response make it hard to intuit a design that can lead to a useful controller. Algorithmic co-generation is a natural solution.

This paper contributes:

- A new probabilistic algorithm for simultaneous design and control (section 3.)
- A comparison showing that the algorithm converges on a known optimal solution (section 5.1.)
- A demonstration that the algorithm beats human design in the motivating problem (section 5.2.)

3 Algorithm

The goal of the algorithm is to find both a design, $\mathcal{D}$, and controller, $\mathcal{C}$, that together maximize the volume of state space over which the controller can successfully stabilize the system. This controller is composed of a set of local controllers associated with both a stable point $\mathbf{x}$ and a a region of attraction of the gains about $\mathbf{x}$. This RoA is represented by an ellipsoidal level set ρ. The closed-loop system will converge to $\mathbf{x}$ from any point within ρ. Since all $\mathbf{x}$ in $\mathcal{C}$ are stable points of the system, any state that lies within any ρ can be successfully stabilized. Thus, the total volume of state space covered by all the ρ in $\mathcal{C}$, $\mathrm{V}(\mathcal{C})$, is a conservative proxy for the amount of state space the system can visit and successfully return to a stable point.

The algorithm for finding the $\mathcal{D}$ and $\mathcal{C}$ that maximizes V($\mathcal{C}$) is shown in Algorithm 1.

Algorithm 1 Control-Volume Based Design Algorithm

$\mathcal{C}$.init(NULL)
best_V $\Leftarrow$ 0
best_design $\Leftarrow$ NULL
while no_convergence **do**
  $\mathcal{D} \Leftarrow$ new parameterized design
  for i = 1 to max_controllers **do**
    $\mathbf{x} \Leftarrow$ random state $\in$ stable states
    [$\mathbf{A}$,$\mathbf{B}$] $\Leftarrow$ linearization of $\mathbf{f}(\mathbf{x},\mathbf{u})$ around $\mathbf{x}$
    [$\mathbf{K}$,$\mathbf{S}$] $\Leftarrow$ LQR($\mathbf{A}$,$\mathbf{B}$,$\mathbf{Q}$,$\mathbf{R}$)
    $\rho \Leftarrow$ level set for the closed loop system using $\mathbf{K}$ about $\mathbf{x}$
    $\mathcal{C}$.add-node($\mathbf{x}$,$\mathbf{K}$,$\mathbf{S}$, ρ)
  end for
  for all node $\mathcal{N}$ in $\mathcal{C}$ **do**
    if $\rho_i \cap \rho \neq \emptyset$ **then**
      $\mathcal{C}$.add-edge(ρ_i, ρ)
    end if
  end for
  V = Volume($\mathcal{C}$)
  if V > best_V **then**
    best_V $\Leftarrow$ V
    best_design $\Leftarrow$ D
  end if
end while

3.1 Algorithm Execution

While the algorithm hasn't exceeded a total number of loops set by `max_iterations` and improvements in the evaluation metric haven't stalled (the number of loops without a better $\mathcal{D}$ hasn't exceeded `max_no_improve`), the algorithm loops between design creation and controller synthesis. First, a new design $\mathcal{D}$ sets the parameters of the system. $\mathcal{D}$ and the independent environment variables define a new dynamical system that needs a controller.

The algorithm populates the controller by first picking a state from the set of stable states, defined either explicitly (with a pre-defined range for a continuous set of points) or implicitly (with a function that checks the stability of each sampled point.) It then linearizes the system about that state and finds the LQR gains for that linearized system. Sum-of-squares finds a verified polynomial level set ρ that is added to the controller along with the LQR gains. This process proceeds until it meets a termination condition like reaching some maximum number of gain nodes.

Once the controller is fully populated with gain nodes, a graph is created by generating an edge between gain nodes. A conservative connection policy will connect only two nodes $\mathcal{N}_1$ and $\mathcal{N}_2$ with $\mathbf{x}_1$ and $\mathbf{x}_2$ such that $\mathbf{x}_1 \in \rho_2$ and $\mathbf{x}_2 \in \rho_1$. The total volume filled by the RoAs in the controller is compared against the largest control volume so far. If the volume of the new controller is larger, its volume becomes the largest control volume with $\mathcal{C}$ and its associated $\mathcal{D}$ becoming the best controller and best design respectively.

In the end, the algorithm produces a design $\mathcal{D}$ and an accompanying controller $\mathcal{C}$ that maximize stabalizable volume.

3.2 Controller Execution

A possible control policy using $\mathcal{C}$ is as follows:

Start with an initial state $\mathbf{x}_0$ and a target state $\mathbf{x}_G$. Find two gain nodes, $\mathcal{N}_0$ and $\mathcal{N}_G$ such that whose level sets encompass $\mathbf{x}_0$ and $\mathbf{x}_G$ respectively. Perform a graph search to connect the two nodes and generate a series of waypoint nodes $\mathcal{N}_1$ ... $\mathcal{N}_k$. Assuming the connection policy in section 3.1, while inside ρ_i the system should use $\mathbf{K}_i$ until it is inside ρ_{i+1} and then switch to $\mathbf{K}_{i+1}$, effectively "hopping" between nodes. This policy provides stability guarantees because the system is always driving itself towards a stable point while within the RoA of that point and using the gains associated with that RoA. Figure 3 shows the RoAs and associated path for a simple example.

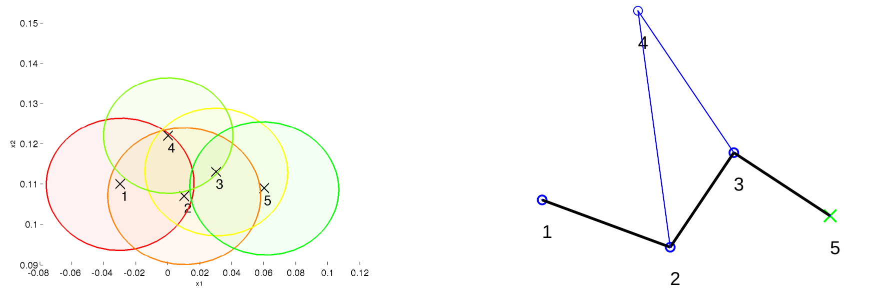
The system can then use other path planners to reach stable states near the node states. One approach is to generate a new sequence of nodes with overlapping RoAs that lead to the goal state, like a fractal of the larger map. This approach assumes smooth dynamics, but the RoA of the node state already makes that assumption.

This control policy is not optimal and leaves significant room for improvement because moving this way gives no performance guarantees beyond stability. It is meant to illustrate how the results of the co-generation algorithm can be incorporated into a control policy that allows the system to traverse volume of $\mathcal{C}$ without additional controller synthesis. Further discussion of control policies is in section 6.2.

4 Implementation Details

The number of gain nodes necessary to allow controllers to cover possibly controllable states varies with the smoothness of the dynamics. $\mathrm{V}(\mathcal{C})$ can converge with a smaller `max_controllers` when the gradients of the dynamics are small and the RoA for individual nodes are large compared to the size of the state space. The user needs to pick the parameters built into the controllers unless they are included as design parameters. In our example the matrices $\mathbf{Q}$ and $\mathbf{R}$ that generate LQR gains are built-in controller parameters.

Different regions of state-space may be more important than others. The algorithm can incorporate this information in two ways. One option is to assign



(a) Two-dimensional regions of attraction. (b) Graph and example control-gain-path from node 1 to node 5.

Fig. 3

more weight towards picking node centers in a region of interest during the sampling step. Another option is adding weight to points in the region of interest during the integration step. Our implementation assigns the same weight to each state.

Several different methods can find a RoA for each gain node. Sum-of-squares is the most widely used and toolboxes like `SPOT` Megretski (2010) simplify the process of generating and manipulating Lyapunov functions. However, SoS does not scale well to high dimensions. Alternatives include simulation-based approaches, DSOS programming and SDSOS programming Ahmadi & Parrilo (2014).

Numerical methods are necessary to find the volume of each controller because there is no good way to find the volume covered by several overlapping ellipses analytically. We use Monte Carlo integration because of the high dimensionality. In our implementation, a set of points from the Halton sequence is generated across the convex hull of the controller and each point checks whether it is contained in any RoA. The Halton sequence's deterministic sampling points allow volumes to be reliably compared while still exhibiting the large dispersion needed to capture high-dimensional volumes. There are many increasingly fancy ways to implement Monte Carlo integrators for more accuracy.

The samples from the integration step also populate the edges of the controller's graph. Each point is already checking whether it is in any RoA, which in the worst case checks each RoA in the controller. To populate the graph, the sample points check whether they are in each RoA, keep track of the RoAs containing them, and then add an edge between all gain nodes associated with those RoAs.

5 Experiments

We demonstrate the co-generation algorithm with two systems: a double pendulum and a robotic inspection spacecraft using novel actuators . The pendulum is a simple example demonstrating that the algorithm converges on a known optimal design. The spacecraft system shows how the co-generation algorithm can generate designs for systems that are intractable for traditional design methods. The generated spacecraft design outperforms the intuitive symmetric design of the type that often outperforms generated designs.

The success criteria for the co-generation algorithm are:

1. Generate a design that can be controlled in a quantifiable volume of state space.

2. Generate a design that is as good or better a intuition-driven human design.

Without criterion 1 the algorithm would be completely useless. Real usefulness comes from criterion 2. Many algorithmic designs succeed at criterion 1 but fail at 2. The human design used as a baseline exhibits the symmetries and even distribution of struts that would intuitively allow the system to stabilize itself in the largest possible state-space volume.

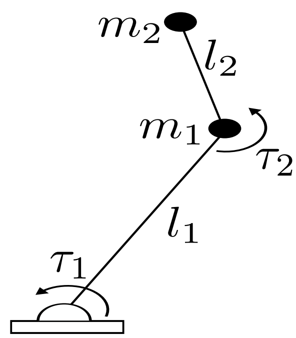


Fig. 4: The pendulum. Co-generation maximizes the stable states achievable by limited τ_1 and τ_2.

5.1 Pendulum

The double pendulum in figure 4 is a simple nonlinear system with four states, two actuators - one at each joint, and an analytical equation of motion. Its simplicity means that its amenable to both hand design and other co-generation methods. Using probabilistc co-generation to find a design and controller is overkill. However, the double pendulum is a good baseline system that demonstrates how probabilistic co-generation will come to the same conclusions as other methods, given enough time.

Equation 2 describes the pendulum's dynamics. For abbreviation, $\alpha = m_1 l_1^2 + m_2(l_1^2 + l_2^2)$, $\beta = m_2 l_2^2$, $\delta = m_2 l_2^2$, $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, $c_{ij} =$

$\cos(\theta_i + \theta_j)$

$$\begin{bmatrix} \alpha + 2\beta c_2 & \delta + \beta c_2 \\ \delta + \beta c_2 & \delta \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -\beta s_2 \dot{\theta}_2 & -\beta s_2 (\dot{\theta}_1 + \dot{\theta}_2) \\ \beta s_2 \dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} m_1 g l_1 c_1 + m_2 g (l_1 c_1 + l_2 c_{12}) \\ m_2 g l_2 c_{12} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{2}$$

Each of the two actuators saturates at $\tau_i = u_{max}$. These two limited actuators create a non-linear system with four continuous sets of stable points and a state-space that is largely unstable.

The double pendulum has a simple design space: the lengths and masses of its arms, l_i and m_i. The arm mass/length design space has a clear optimum design using the controllable volume metric: minimizing the masses and lengths of the arms gives the actuators maximum control over the pendulum's joint angles. Optimization via MATLAB's `fmincon` confirms the optimality of this design, making it the clear choice as an experimental baseline. The volume of the interconnected RoAs for this design provides a baseline controllable volume.

In figure 5, the co-generation algorithm generates designs using increasing values of `max_iterations`. The generated design's stabilizable volume approaches the baseline value as `max_iterations` increases, demonstrating convergence on the optimal design.

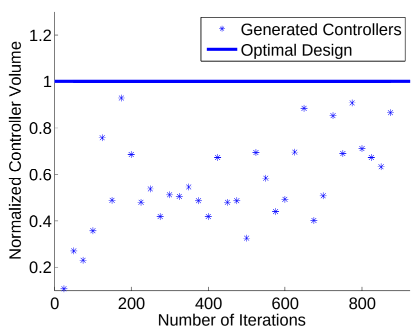


Fig. 5: The controllable volume of generated designs converges on the controllable volume of the optimal design as the algorithm's iterations increase. The volumes are normalized by the controllable volume of the optimal design.

5.2 Spacecraft

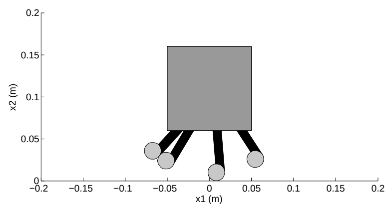
The other test system, an inspection spacecraft, is a complex system that uses novel electromagnetic actuators to interact with a conductive target without contact. The control forces from these actuators are nonlinear, position dependent, and saturate. Coupled with the spacecraft's rigid-body dynamics, the actuators create a system that exhibits all the criteria listed in section 2. This complexity causes traditional methods to fall short and motivates the creation of the co-generation algorithm.

The experimental model in figure 1 is a simplified version of a real system currently in development Reinhardt & Peck (2014). The model spacecraft is a rigid body with three degrees of freedom (two translational, one rotational.) Struts attach the actuators to the main mass of the system so that the actuators' positions aren't constrained by the inspector's body geometry. The angles between the spacecraft body and the four struts are the free parameters, $\mathbf{p}$, specified during the design of the system. The length of the struts and mass of the spacecraft are fixed parameters, while the position of the surface (that magnetically interacts with the actuators) and dynamics of the actuators are external environmental variables.

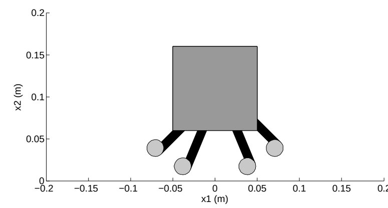
Optimization methods fail to generate designs and controllers for this system, even with a smaller set of parameters than the real system. Both `fmincon` and `SNOPT` can't provide a baseline design because they fail to converge on solutions. This failure was what drove our new method in the first place. Without optimized designs, a system designer would need to turn to their intuition. The human baseline is this intuitive design that places struts symmetrically spanning the space of possible positions, which ideally would make the system stabilizable in as many configurations as possible.

The results in figure 6 show the design and a projection of the control volume for both the human and algorithmically generated design. The designs (subfigures 6a and 6b) show the body, strut positions, and target surface. Subfigures 6c and 6d) show 2-d projections of the regions of attraction onto the x-y plane. The baseline design has a stabilizable volume $V_h = 4.8E - 11$. After 65 iterations, the algorithm produced a design with a stabilizable volume $V_a = 1.8E - 10$. The ratio $\frac{V_a}{V_h} = 3.75$ shows that the algorithm succeeds on both criteria 1 and 2.

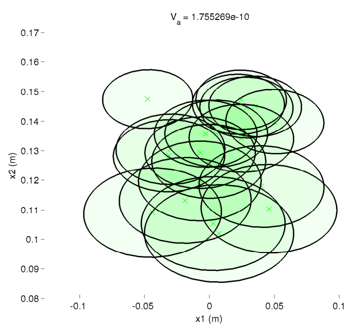
Closer inspection of the resulting design yields unexpected insight into the system as well. The algorithm's design consistently places a strut pointed straight down; its actuator as close to the surface as possible. The state-dependence of the actuators explains this placement because their effectiveness decreases with r^4 from the surface. However, there is a trade-off between this arrangement and the human-designed baseline which places actuators farther out to the side to provide more control when the inspector rotates. Post hoc this trade-off is obvious and could be analyzed with traditional methods. However, the size of the design space and complexity of the dynamics completely obscures that clarity during the initial design process. This insight is specific to the particular model, but it demonstrates how the co-generation algorithm can lead to insights about complex systems. The co-generation leads to both an expanded control volume and a design insight that would have been lost in a traditional design-and-then-controller-synthesis process.



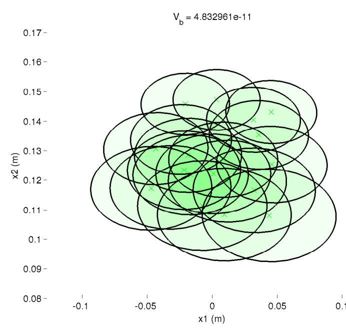
(a) Algorithmic Design



(b) Baseline Design



(c) Algorithmic Controller



(d) Baseline Controller

Fig. 6: Algorithmic design and controllers (left) and human design and controllers (right.) The system comprises a main body attached to a set of nonlinear actuators by struts. The co-generation algorithm uses the angles of these struts as the free parameters ($\mathbf{p}$). In an unexpected result, the algorithm showed that symmetry between the four actuators led to less controllable area than a design with a central actuator.

6 Discussion

The algorithm presented here is meant as a proof-of-concept for algorithmic design based on a controller metric. There are many possible extensions and areas for future work both in the design and controller halves.

6.1 Design

There are multiple methods for generating a design at each iteration. The implementation in this paper uses a naive approach: a new set of parameters is selected randomly each iteration. Two other possibilities for design generation are genetic algorithms (GA) and optimization. GAs are traditionally used for algorithmic design generation and would be a good fit. However, they require significant tuning and without it, don't outperform random generation. An optimization method using the control volume as a cost function could replace the entire co-generation algorithm if the number of parameters is small and the dynamics smoothly change as parameters those parameters change. Neither of these requirements are guaranteed, especially if some parameters can change discretely (three vs. four struts.)

6.2 Control

New center points for new gain nodes can be generated in different ways to maximize coverage or connections between nodes. Selecting only new center points that fall within an existing RoA will maximize connections between nodes; the new node's RoA is guaranteed to intersect the existing RoA, forming a connection. Selecting gain node centers that are already within the RoA of another node guarantees that the control policy described in section 3.2 will succeed. Alternately, selecting only new center points that do not fall within, an existing region will increase coverage and decrease redundant controllers for the same points. Dynamics-based distance metrics can also be used to pick new center points that are 'dynamically close' to existing regions, leading to new regions that expand coverage and are likely to connect to others Perez et al. (2012); Glassman & Tedrake (2010). $V(\mathcal{C})$ need not necessarily be composed of many LQR regions of attraction. If the dynamics are analytically tractable, SoS can find a single parameterized RoA for the entire range of goal states Majumdar (2012). This single RoA reduces computation significantly because it requires only one verification but it doesn't lead directly to controller synthesis as a graph of connected gain nodes.

Time invariant LQR (TILQR) controllers must be centered around a stable point. This restricts $\mathcal{C}$ to a volume of state space immediately around the set of goal points while using TILQR gains. Limiting the controller to TILQR gains limits $V(\mathcal{C})$, trading volume for consistency. $\mathcal{C}$ could expand farther into state space by including Time-Varying LQR (TVLQR) gains through methods like LQR trees Tedrake et al. (2010). The trees' TVLQR controllers would stabilize the system to pre-computed trajectories that lead to stable points or other time-varying LQR controllers. LQR trees have verified RoAsthat can add to the $V(\mathcal{C})$ design metric. The downside of the expanded control volume is that LQR trees depend on accurate trajectory optimization which can fail to converge for complicated systems, especially when the specific dynamics are unknown a priori because of the design's mutability.

The final control policy need not be limited to jumping between the lqr gains in $\mathcal{C}$. Several different robust on-line planners exist, including Model-Predictive Control (MPC) and LQR trees. These will outperform LQR gain hopping most of the time. When using these controllers, $\mathcal{C}$ remains valuable because it can provide constraints for MPC that prevent it from driving the system to an uncontrollable state. Additionally, the gain nodes could provide goal regions for LQR tree generation.

7 Summary and Conclusion

Designs for complex dynamical systems often require decisions that can benefit from a tight loop between the design parameters and the control system instead of the traditional approach: design first, controller later. This paper presents an approach to algorithmic design that uses stablizable volume of state space as a metric. This volume acts as a conservative metric for the robustness of the generated controller and by proxy, the quality of the design. The approach is appropriate for high-dimensional systems with several parameters thanks to its probabilistic nature.

References

A. A. Ahmadi & P. a. Parrilo (2014). 'Towards Scalable Algorithms with Formal Guarantees for Lyapunov Analysis of Control Systems via Algebraic Optimization'. *Domino.Research.Ibm.Com* **25479**:11.

J. E. Auerbach & J. C. Bongard (2010). 'Dynamic Resolution in the Co-Evolution of Morphology and Control'. *Artificial Life XII. Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems* pp. 451–458.

S. S. Bahakim & L. a. Ricardez-Sandoval (2014). 'Simultaneous design and MPC-based control for dynamic systems under uncertainty: A stochastic approach'. *Computers and Chemical Engineering* **63**:66–81.

V. Bansal, et al. (2000). 'Simultaneous Design and Control Optimisation Under Uncertainty'. *Computers and Chemical Engineering* **24**:261–266.

R. R. Burridge, et al. (1999). 'Sequential Composition of Dynamically Dexterous Robot Behaviors'. *The International Journal of Robotics Research* **18**(6):534–555.

H. Choset, et al. (1999). 'Path planning and control for AERCam, a free-flying inspection robot in space'. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)* **2**(2):74–81.

D. Conner, et al. (2003). 'Composition of local potential functions for global robot control and navigation'. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)* **4**(October).

D. Conner, et al. (2006). 'Towards provable navigation and control of nonholonomically constrained convex-bodied systems'. *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA06)* (May):4416–4418.

E. Glassman & R. Tedrake (2010). 'A quadratic regulator-based heuristic for rapidly exploring state space'. In *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5021–5028.

J. Gowdy & a.a. Rizzi (1999). 'Programming in the architecture for agile assembly'. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)* **4**.

a. L. Hale, et al. (1985). 'Optimal simultaneous structural and control design of maneuvering flexible spacecraft'. *Journal of Guidance, Control, and Dynamics* **8**(1):86–93.

G. S. Hornby, et al. (2003). 'Generative representations for the automated design of modular physical robots'. *IEEE Transactions on Robotics and Automation* **19**(4):703–719.

J. W. Hurst, et al. (2007). 'Design and Philosophy of the {B}i{MASC}, a Higly Dynamic Biped'. *Proceedings of the IEEE International Conference of Robotics and Automation* pp. 1863–1868.

R. Kajiwara, I.; Haftka (1999). 'Simultaneous Optimum Design of Shape and Control System for Micro Air Vehicles'. In *40TH STRUCTURES, STRUCTURAL DYNAMICS, AND MATERIALS CONFERENCE AND EXHIBIT*.

S. Karaman & E. Frazzoli (2013). 'Sampling-based optimal motion planning for non-holonomic dynamical systems'. *2013 IEEE International Conference on Robotics and Automation* pp. 5041–5047.

L. Kavraki, et al. (1996). 'Probabilistic roadmaps for path planning in high-dimensionalconfiguration spaces'. *Robotics and Automation, IEEE Transactions on* **12**(4):566 – 580.

D. Koditschek (1989). 'The application of total energy as a Lyapunov function for mechanical control systems'. In *Dynamics and control of multibody systems*, pp. 131–156. American Mathematical Soc.

I. K. Kookos & J. D. Perkins (2001). 'An Algorithm for Simultaneous Process Design and Control'. *Industrial & Engineering Chemistry Research* **40**:4079–4088.

K. Kotay, et al. (1998). 'The self-reconfiguring robotic molecule: Design and control algorithms'. In *Workshop on Algorithmic Foundations of Robotics*, pp. 376–386. Citeseer.

H. Kwakernaak & R. Sivan (1972). *Linear optimal control systems*, vol. 1. Wiley-Interscience New York.

S. LaValle & J. Kuffner, J.J. (2001). 'Randomized kinodynamic planning'. *International Journal of Robotics Research* **20**(5):378–400.

T. Lozano-Perez, et al. (1984). 'Automatic Synthesis of Fine-Motion Strategies for Robots'. *The International Journal of Robotics Research* **3**:3–24.

A. Majumdar (2012). 'Algebraic verification for parameterized motion planning libraries'. In *American Control Conference*, pp. 250–257.

M. T. Mason (1985). 'The Mechanics of Manipulation'. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on (Volume:2)*, pp. 544–548.

A. Megretski (2010). 'Systems polynomial optimization tools (SPOT)'.

I. Mitchell, et al. (2005). 'A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games'. *IEEE Transactions on Automatic Control* **50**(7):947–957.

S. Papachristodoulou, Antonis; Prajna (2002). 'the Sum of Squares Decomposition ''. In *2002 Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, pp. 3482–3487.

A. Perez, et al. (2012). 'LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics'. In *Proceedings - IEEE International Conference on Robotics and Automation*, no. 0, pp. 2537–2542.

J. Rastegar, et al. (1999). 'Task-specific optimal simultaneous kinematic, dynamic, and control design of high-performance robotic systems'. *IEEE/ASME Transactions on Mechatronics* **4**(4):387–395.

T. Ravichandran, et al. (2006). 'Simultaneous plant-controller design optimization of a two-link planar manipulator'. *Mechatronics* **16**:233–242.

B. Reinhardt & M. Peck (2014). 'A New Actuator for On-Orbit Inspection'. In *Proceedings of the AIAA/USU Conference on Small Satellites*.

B. Reinhardt & M. Peck (2015). 'A New Actuator for On-Orbit Inspection'. In *International Conference for Robotics and Automation*, p. To Appear.

P. Reist & R. Tedrake (2010). 'Simulation-based LQR-trees with input and state constraints'. *Proceedings - IEEE International Conference on Robotics and Automation* pp. 5504–5510.

L. a. Ricardez-Sandoval, et al. (2009). 'Application of Robust Control Tools to the Simultaneous Design and Control of Dynamic Systems'. *Industrial & Engineering Chemistry Research* **48**:801–813.

L. a. Ricardez-Sandoval, et al. (2010). 'Simultaneous design and control: A new approach and comparisons with existing methodologies'. *Industrial and Engineering Chemistry Research* **49**:2822–2833.

P. Scokaert & J. Rawlings (1998). 'Constrained linear quadratic regulation'. *IEEE Transactions on Automatic Control* **43**(8):1163–1169.

J. P. Shoer & M. a. Peck (2013). 'Flux-Pinned Interfaces for the Assembly, Manipulation, and Reconfiguration of Modular Space Systems'. *The Journal of the Astronautical Sciences* **57**(3):667–688.

J. Sobieszczanski-Sobieski & R. T. Haftka (1997). 'Multidisciplinary aerospace design optimization: survey of recent developments'. *Structural Optimization* **14**(Berkes 1990):1–23.

R. Tedrake, et al. (2010). 'LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification'. *The International Journal of Robotics Research* **29**:1038–1052.

M. M. Tobenkin, et al. (2011). 'Invariant funnels around trajectories using Sum-of-Squares programming'. *IFAC Proceedings Volumes (IFAC-PapersOnline)* **18**(1):9218–9223.

X. Yao & T. Higuchi (1999). 'Promises and challenges of evolvable hardware'. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **29**(1):87–97.

M. Yim, et al. (2007). 'Modular Self-reconfigurable Robot Systems: Challenges and Opportunities for the Future'. {*IEEE*} *Robotics \& Automation Magazine* **14**(March):43–52.