Alternative Control Trajectory Representation for the Approximate Convex Optimization of Non-Convex Discrete Energy Systems

Eric M. Burger, Scott J. Moura

Energy, Control, and Applications Lab, University of California, Berkeley

Abstract

Energy systems (e.g. ventilation fans, refrigerators, and electrical vehicle chargers) often have binary or discrete states due to hardware limitations and efficiency characteristics. Typically, such systems have additional programmatic constraints, such as minimum dwell times to prevent short cycling. As a result, non-convex techniques, like dynamic programming, are generally required for optimization. Recognizing developments in the field of distributed convex optimization and the potential for energy systems to participate in ancillary power system services, it is advantageous to develop convex techniques for the approximate optimization of energy systems. In this manuscript, we develop the alternative control trajectory representation – a novel approach for representing the control of a non-convex discrete system as a convex program. The resulting convex program provides a solution that can be interpreted stochastically for implementation.

Keywords: alternative control trajectory, discrete systems, convex optimization, distributed optimization

1. Background and Motivation

A fundamental requirement of the electric power system is to maintain a continuous and instantaneous balance between generation and load. The variability of renewable energy resources, particularly wind and solar, poses a challenge for power system operators. Namely, as renewable penetration increases, it will be necessary for operators to procure more ancillary services, such as regulation and load following, to maintain balance between generation and load [1][2]. In the long-term, grid-scale storage technologies (e.g. flywheels, batteries, etc.) are sure to play a major role in providing these ancillary services [3][4]. In the near-term, there is a high potential for aggregated loads, in particular electric vehicles (EVs) and thermostatically controlled loads (TCLs), to providing such ancillary services [5][6][7][8].

The advantages of responsive aggregated loads over large storage technologies include: 1) they are distributed throughout the power system thus providing spatially and temporally distributed actuation; 2) they employ simple and fast local actuation well-suited for real-time control; 3) they are robust to outages of individuals in the population; and 4) they, on the aggregate, can produce a quasi-continuous response despite the discrete nature of the individual controls [6][9][10].

Energy systems like EVs and TCLs often have binary or discrete states due to hardware limitations and efficiency characteristics. Consequently, non-convex techniques are generally required for optimal control. This poses a challenge for load aggregation applications since distributed optimization methods generally require linearity or convexity in the agents. In this manuscript, we develop the alternative control trajectory representation – a novel approach for representing the control of a non-convex discrete system as a convex program. This representation enables the approximate optimization of energy systems using distributed convex algorithms, such as the alternating direction

*Corresponding author: Eric M. Burger,
Email: ericburger@berkeley.edu
**Affiliation Address: Energy, Control, and Applications Lab (eCAL), 611 Davis Hall, Department of Civil and Environmental Engineering, University of California, Berkeley, Berkeley, CA 94720, USA.

method of multipliers (ADMM), and provides a solution that can be interpreted stochastically for implementation.

This paper is organized as follows. Section 2 briefly describes the optimization of non-convex systems and section 3 presents the alternative control trajectory representation. Section 4 overviews the incorporation of the ACT representation into a convex program and the stochastic interpretation of the solution. Section 5 describes the incorporation of the ACT representation into a distributed optimization algorithm, the statistical characteristics of the solution, and an iterative method for reducing variance by inducing sparsity. Finally, Section 6 provides an illustrative example of the proposed modeling and optimization approach.

2. Optimization of Non-Convex Systems

In this section, we consider the optimization of an arbitrary discrete-time system represented by the state-space model

$$\begin{aligned} x^k &= G(x^{k-1}, u^k) \\ y^k &= H(x^k, u^k) \end{aligned} \tag{1}$$

where G and H are known functions, x^k is the state of the system, u^k is the exogenous input, y^k is the output, and k denotes the integer-valued time step. For simplicity, this paper will only consider the univariate case (i.e x^k, u^k, and y^k are univariate, $G : \mathbf{R}^2 \to \mathbf{R}$, and $H : \mathbf{R}^2 \to \mathbf{R}$). Functions G and H may be any closed deterministic function (i.e. non-convex, piece-wise, semi-continuous, etc.) and x^k, u^k, and y^k may be continuous or discrete.

We would like to solve an optimization problem of the form

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F_0(y) \\ \text{subject to} \quad & F_i(x, u) \le b_i, \; i = 1, \ldots, M \\ & x^k = G(x^{k-1}, u^k), \; k = 1, \ldots, N \\ & y^k = H(x^k, u^k), \; k = 1, \ldots, N \\ & x^0 = x_0 \end{aligned} \tag{2}$$

where $F_0 : \mathbf{R}^N \to (-\infty, \infty]$ is a closed convex objective function, N is the number of time steps, and x_0 is the initial state. Functions $F_i : \mathbf{R}^2 \to \mathbf{R}$, $i = 1, \ldots, M$ represent the constraints of the system. Like G and H, F_i may be any closed deterministic function.

There are a number of non-convex optimization techniques, such as dynamic programming and genetic algorithms, suitable for solving (2) to identify a control trajectory u^* that optimizes the system. Convex optimization techniques, however, are unsuitable given the non-convex constraints and the discrete states, inputs, and outputs.

3. Alternative Control Trajectory Representation

In this section, we introduce the alternative control trajectory (ACT) representation, a novel approach for representing the control of non-convex systems in a manner suitable for convex programming. Put simply, we simulate the system under multiple alternative control inputs in order to generate a discrete set of output trajectories. These alternative control trajectories can be incorporated into a convex program as a linear constraint, thereby enforcing feasibility. By solving the convex program, we produce a solution that can be interpreted stochastically for implementation.

It should be noted that the alternative control trajectories do not represent the full decision space of the original optimization program (2) and that the stochastic solution has no optimality guarantee. Rather, the contribution of the ACT representation is to enable the optimization of a large population of non-convex agents using distributed convex optimization. Accordingly, by employing the ACT representation, we are accepting suboptimality in the individual objectives in order to achieve optimality in the global objective.

To produce the alternative control trajectory representation of a system, we first define N_a input trajectories for N_t time steps

$$\begin{gathered} u_j = (u_j^1, u_j^2, \ldots, u_j^{N_t}) \\ \forall \; j = 1, \ldots, N_a \end{gathered} \tag{3}$$

with variable $u_j \in \mathbf{R}^{N_t}$ and $u_j^k \in \mathbf{S}_u$ for $k = 1, ..., N_t$, where $\mathbf{S}_u$ is the discrete or continuous constraint set of feasible inputs. Each of the input trajectories u_j must be distinct and should be selected to produce a distinguishable change in the system's output (i.e. performance extremes, efficiency optimum, etc.). Regardless of whether the input is discrete or continuous, the set of alternative input trajectories express only a small but key portion of the true decision space.

Next, for each input trajectory u_j, we simulate the system model (1) according to the update function G with $x^0 = x_i$ while imposing any additional constraints (represented by H_i in (2)). Given the simulation results, we generate N_a alternative state and output trajectories as defined by the x_j and y_j, respectively.

$$\begin{aligned} x_j &= (x_j^1, x_j^2, \ldots, x_j^{N_t}) \\ y_j &= (y_j^1, y_j^2, \ldots, y_j^{N_t}) \\ &\forall\ j = 1, \ldots, N_a \end{aligned} \tag{4}$$

The input, state, and output trajectories can be expressed compactly as

$$\begin{aligned} \mathbf{U} &= (u_1, u_2, \ldots, u_{N_a}) \\ \mathbf{X} &= (x_1, x_2, \ldots, x_{N_a}) \\ \mathbf{Y} &= (y_1, y_2, \ldots, y_{N_a}) \end{aligned} \tag{5}$$

with variables $\mathbf{U}$, $\mathbf{X}$, and $\mathbf{Y}$ representing the set of all u_j, x_j, and y_j sets for $j = 1, \ldots, N_a$. Naturally, we can also view $\mathbf{U}$, $\mathbf{X}$, and $\mathbf{Y}$ as matrices $\in \mathbf{R}^{N_a \times N_t}$ such that the rows represent the alternative trajectories and the columns represent the time step k.

In the case that functions G and/or H are not injective/one-to-one and the distinctness of u_j does not guarantee the distinctness of x_j or y_j, it is necessary to reduce the number of trajectories in $\mathbf{U}$, $\mathbf{X}$, and $\mathbf{Y}$. We define the number of *distinct* alternative control trajectories as N_d such that $N_d \in \{1, \ldots, N_a\}$.

4. Convex Optimization

In this section, we detail how the ACT representation described above can be introduced into a convex program. To begin, we introduce a variable $w \in \{0,1\}^{N_d}$ such that

$$w_j = \begin{cases} 1 & \text{if trajectory } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ j = 1, \ldots, N_d \tag{6}$$

Thus, if $j = 1$ is the selected trajectory (i.e. $w_1 = 1$)

$$\begin{aligned} \mathbf{U}^T w &= u_1 \\ \mathbf{X}^T w &= x_1 \\ \mathbf{Y}^T w &= y_1 \end{aligned}$$

The integer/binary program below demonstrates how $\mathbf{Y}$ and w can be introduced to solve for the optimal trajectory

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & F(\mathbf{Y}^T w) \\ \text{subject to} \quad & \textstyle\sum w_j = 1 \\ & w \in \{0,1\}^{N_d} \end{aligned} \tag{7}$$

where $F : \mathbf{R}^{N_t} \to (-\infty, \infty]$ is a closed convex objective function. The above program is an example of the generalized assignment problem (GAP). If feasible, (7) guarantees that only one component of minimizer w^* is non-zero. Therefore, $y^* = \mathbf{Y}^T w^*$ is the optimal output trajectory within the discrete set defined by $\mathbf{Y}$. However, the binary constraint makes the program non-convex and NP-complete. By relaxing the binary constraint such that $\hat{w} \in \mathbf{R}^{N_d}$, we can express the convex program as

$$\begin{aligned} \underset{\hat{w}}{\text{minimize}} \quad & F(\mathbf{Y}^T \hat{w}) \\ \text{subject to} \quad & \textstyle\sum \hat{w}_j = 1 \\ & \hat{w} \geq 0 \\ & \hat{w} \in \mathbf{R}^{N_d} \end{aligned} \tag{8}$$

The program is now convex and the decision variable continuous. By minimizing the objective function with respect to $\hat{w}$, we allow the convex program to form weighted averages of the alternative output trajectories. Therefore, $\hat{y}^* = \mathbf{Y}^T \hat{w}^*$ is the optimal weighted average of the output trajectories within the discrete set defined by $\mathbf{Y}$. However, for many systems, the solution defined by $\hat{w}^*$ is not realizable (e.g. $\hat{u}^* = \mathbf{U}^T \hat{w}^*$ is not within the feasible space, $\hat{y}^{*,k} \neq H(\hat{x}^{*,k}, \hat{u}^{*,k})$, etc.). To produce a realizable solution, we can interpret $\hat{w}^*$ stochastically, as described in the next section.

4.1. Stochastic Solution

Due to the linear constraints, the optimal solution $\hat{w}_j^*$ is $\in [0,1]$ for $j = 1, \ldots, N_d$ and in practice, $\hat{w}_j^*$ can be interpreted as the probability of selecting control trajectory j. Thus, we can implement a single trajectory $\tilde{y} \in \mathbf{Y}$ based on the discrete probability distribution $\hat{w}^*$. Expressed alternatively, we can generate a discrete random variable $W \in \{1, \ldots, N_d\}$ such that $\hat{w}_j^* = \Pr(W = j)$ for $j = 1, \ldots, N_d$. The value of W represents the index of the stochastically selected control trajectory. Thus, we can define a variable $\tilde{w} \in \{0,1\}^{N_d}$, representing the stochastic solution of (8), as

$$\tilde{w}_j = \begin{cases} 1 & \text{if } W = j \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$
$$\forall\ j = 1, \ldots, N_d$$

The selected output trajectory is therefore given by $\tilde{y} = \mathbf{Y}^T \tilde{w}$. By treating $\hat{w}^*$ as a discrete probability distribution, $\hat{y}^*$ becomes the probability-weighted average of possible output trajectories (as defined by $\mathbf{Y}$). Therefore, $\hat{y}^*$ is the expected value of $\tilde{y}$.

$$\mathrm{E}[\tilde{y}] = \hat{y}^* \qquad (10)$$

To summarize, the optimal solution to (7) is physically realizable (i.e. only one component of w^* is non-zero) but not solvable using convex optimization. By contrast, (8) is convex but the optimal solution is not realizable (i.e. all components of $\hat{w}^*$ may be non-zero). Using (9), we can transform $\hat{w}^*$ into $\tilde{w}$, which is realizable (i.e. only one component of $\tilde{w}$ is non-zero). It should be noted that w^* and $\hat{w}^*$ are guaranteed to be optimal solutions to (7) and (8), respectively. However, $\tilde{w}$ may be an optimal or sub-optimal solution to both (7) and (8).

Throughout this paper, we refer to the optimal output trajectory ($y = \mathbf{Y}^T w$) produced by (7) as the *discrete* solution y^* ($w^* \in \{0,1\}^{N_d}$), by (8) as the *continuous* solution $\hat{y}^*$ ($\hat{w}^* \in \mathbf{R}^{N_d}$), and by (8) and (9) as the *stochastic* solution $\tilde{y}$ ($\tilde{w} \in \{0,1\}^{N_d}$). It should be noted that y^* and $\hat{y}^*$ are deterministic whereas $\tilde{y}$ is, of course, stochastic.

5. Distributed Optimization

In this manuscript, we have detailed the ACT representation for expressing the control of a non-convex discrete system as a convex program and have discussed how the solution can be interpreted stochastically for implementation. In this section, we briefly discuss the application of this approach to distributed convex optimization.

Consider the generic *sharing* problem of the form

$$\underset{y}{\text{minimize}} \quad \sum f_i(y_i) + g(\sum y_i) \qquad (11)$$

with variables $y_i \in \mathbf{S}_i^{N_y}$, the decision variable of agent i for $i = 1, \ldots, N_p$, where $\mathbf{S}_i$ represents the convex constraint set of agent i, N_p the number of agents in the population, N_y is the length of y_i, f_i is the convex objective function for agent i, and g is the shared convex objective function of the population. The function g takes as input the sum of the individual agent's decision variables, y_i. The sharing problem allows each agent in the population to minimize its individual/private cost $f_i(y_i)$ as well as the shared objective $g(\sum y_i)$. The problem is known to be solvable using iterative methods of distributed convex optimization, such as the alternating direction of multipliers algorithm (ADMM) [11].

The ACT representation can be incorporated into the objective functions of (11) as given by

$$\begin{array}{ll} \underset{\hat{w}}{\text{minimize}} & \sum f_i(\mathbf{Y}_i^T \hat{w}_i) + g(\sum \mathbf{Y}_i^T \hat{w}_i) \\ \text{subject to} & \sum \hat{w}_{i,j} = 1 \\ & \hat{w}_i \geq 0 \\ & \hat{w}_i \in \mathbf{R}^{N_{d,i}} \\ & \forall\ i = 1, \ldots, N_p \end{array} \qquad (12)$$

with variables $\hat{w}_i \in \mathbf{R}^{N_{d,i}}$, the decision variable of agent i, $\mathbf{Y}_i \in \mathbf{R}^{N_{d,i} \times N_y}$, the set of alternative output trajectories for agent i, and $N_{d,i}$, the number of distinct trajectories in $\mathbf{Y}_i$ for $i = 1, \ldots, N_p$. Because the objective function and constraints of each agent are separable, the problem can be solved in a distributed manner. The optimal output $\hat{y}_i^* = \mathbf{Y}_i^T \hat{w}_i^*$ for $i = 1, \ldots, N_p$ is the *continuous* solution of each agent in the population. Thus, $\tilde{y}_i = \mathbf{Y}_i^T \tilde{w}_i$ is the final *stochastic* solution and can be implemented by each agent.

5.1. Aggregated Stochastic Solution

When trying to optimize the behavior of a population, we are interested in understanding the relationship between the aggregate of the continuous and stochastic solutions, as given by

$$\begin{aligned} \hat{S} &= \sum \hat{y}_i^* \\ \tilde{S} &= \sum \tilde{y}_i \\ e &= \tilde{S} - \hat{S} \end{aligned} \qquad (13)$$

with variables $\hat{S} \in \mathbf{R}^{N_y}$, the sum of the continuous solutions, $\tilde{S} \in \mathbf{R}^{N_y}$, the sum of the stochastic solutions, and $e \in \mathbf{R}^{N_y}$, the error between $\hat{S}$ and $\tilde{S}$ (ideally, $e \in \{0\}^{N_y}$).

Because $\hat{y}_i^*$ is the expected value of $\tilde{y}_i$, $\hat{S}$ is the expect value of $\tilde{S}$

$$\begin{aligned}\mathrm{E}[\tilde{S}] &= \sum \mathrm{E}[\tilde{y}_i] \\ &= \sum \hat{y}_i^* \\ &= \hat{S}\end{aligned} \tag{14}$$

The error e is therefore related to the variance of $\tilde{S}$, given by

$$\begin{aligned}\mathrm{Var}(\tilde{S}) &= \mathrm{E}[(\tilde{S} - \mathrm{E}[\tilde{S}])^2] \\ &= \mathrm{E}[(\tilde{y}_1 - \hat{y}_1^* + \ldots + \tilde{y}_{N_p} - \hat{y}_{N_p}^*)^2] \\ &= \mathrm{E}[(\tilde{y}_1 + \ldots + \tilde{y}_{N_p})^2] \\ &\quad - (\hat{y}_1^* + \ldots + \hat{y}_{N_p}^*)^2 \\ &= \sum_{i=1}^{N_p} (\mathrm{E}[\tilde{y}_i^2] - (\hat{y}_i^*)^2) \\ &\quad + \sum_{i \neq j} (\mathrm{E}[\tilde{y}_i]\mathrm{E}[\tilde{y}_j] - (\hat{y}_i^* \hat{y}_j^*)) \\ &= \sum_{i=1}^{N_p} \mathrm{Var}(\tilde{y}_i) + \sum_{i \neq j} \mathrm{Cov}(\tilde{y}_i, \tilde{y}_j)\end{aligned} \tag{15}$$

Because the random variables are uncorrelated ($\mathrm{Cov}(\tilde{y}_i, \tilde{y}_j) = 0, \forall (i \neq j)$), the variance of $\tilde{S}$ reduces to

$$\begin{aligned}\mathrm{Var}(\tilde{S}) &= \sum_{i=1}^{N_p} \mathrm{Var}(\tilde{y}_i) \\ &= \sum_{i=1}^{N_p} (\mathrm{E}[\tilde{y}_i^2] - (\hat{y}_i^*)^2)\end{aligned} \tag{16}$$

Since $\hat{w}^*$ is a discrete probability distribution

$$\begin{aligned}\mathrm{Var}(\tilde{S}) &= \sum_{i=1}^{N_p} \sum_{j=1}^{N_{d,i}} \hat{w}_{i,j}^* (y_{i,j} - \hat{y}_i^*)^2 \\ &= \sum_{i=1}^{N_p} \left(\sum_{j=1}^{N_{d,i}} (\hat{w}_{i,j}^* y_{i,j}^2) - (\hat{y}_i^*)^2 \right) \\ &= \sum_{i=1}^{N_p} \sum_{j=1}^{N_{d,i}} (\hat{w}_{i,j}^* y_{i,j}^2) - \sum_{i=1}^{N_p} (\hat{y}_i^*)^2\end{aligned} \tag{17}$$

where variable $y_{i,j}$ is the j-th alternative output trajectory for agent i.

In the remainder of this section, we discuss two particular characteristics that impact the error $e = \tilde{S} - \hat{S}$ and the variance of $\tilde{S}$: the homogeneity/heterogeneity of the agents in the population and the sparsity of the discrete probability distribution $\hat{w}_i^*$ (i.e. the number of non-zero terms) for $i = 1, \ldots, N_p$.

For a population of highly homogeneous agents with identical output trajectories and objective functions, solving (12) will cause each agent to converge to the same solution $\hat{w}_i^*$. Effectively, the output of each agent is defined by the same random variable $\tilde{y}_i$ with the same probability distribution $\hat{w}_i^*$ and expected value $\hat{y}_i^*$. This is a special case where

$$\begin{aligned}\mathrm{E}(\tilde{S}) &= N_p \hat{y}_i^* \\ \mathrm{Var}(\tilde{S}) &= N_p \sum_{j=1}^{N_{d,i}} \hat{w}_{i,j}^* (y_{i,j} - \hat{y}_i^*)^2\end{aligned} \tag{18}$$

and the probability mass of $\tilde{S}$ becomes more and more concentrated about $\mathrm{E}(\tilde{S}) = \hat{S}$ as the number of agents N_p increases. If N_p is very large, the distribution has a narrow peak at $\hat{S}$ regardless of the sparsity of $\hat{w}_i^*$. Therefore, by the law of large numbers, $\tilde{S} \to \hat{S}$ and $e \to \{0\}^{N_y}$ as $N_d \to \infty$. As the heterogeneity of the population increases, this characteristic weakens as the probability mass of $\tilde{S}$ flattens. For a heterogeneous population, the output of each agent is no longer defined by the same random variable and (12) is less likely to converge to similar probability distributions.

The sparsity of $\hat{w}_i^*$ also impacts the variance of $\tilde{y}_i$. In the most sparse case, only one term in $\hat{w}_i^*$ is non-zero for every agent in the population. Therefore, $\tilde{y}_i$ is a constant random variable ($\mathrm{Var}(\tilde{y}_i) = \{0\}^{N_y}$) equal to its expected value ($\tilde{y}_i = \hat{y}_i^*$). Accordingly, $\mathrm{Var}(\tilde{S}) = \{0\}^{N_y}$ and $\tilde{S} \to \hat{S}$.

In the least sparse case, every agent is equally likely to implement any one of its control trajectories (i.e. $\hat{w}_{i,j}^* = 1/N_{d,i} \; \forall \; j = 1, \ldots, N_{d,i}$). Thus

$$\begin{aligned}\mathrm{E}(\tilde{S}) &= \hat{S} \\ \mathrm{Var}(\tilde{S}) &= \sum_{i=1}^{N_p} \sum_{j=1}^{N_{d,i}} \frac{(y_{i,j} - \hat{y}_i^*)^2}{N_{d,i}}\end{aligned} \tag{19}$$

and the aggregate behavior of the population becomes highly stochastic, especially as heterogeneity in $\mathbf{Y}_i$ increases.

5.2. Inducing Sparsity

The stochasticity of $\tilde{S}$ diminishes our ability to optimally control the behaviour of the distributed population. Particularly, in order to optimize a highly heterogeneous population, it would be desirable to force the variance of $\tilde{S}$ towards zero. In this case, we would no longer rely on the law of large numbers to drive $\tilde{S}$ towards the expected value $\hat{S}$.

To decrease the variance, we focus on inducing sparsity in the continuous solution $\hat{w}^*$ of a single system. In this section, we begin by discussing the challenges of inducing sparsity and conclude with an iterative optimization technique. This iterative technique adds a linear cost function to (8) which drives the terms in $\hat{w}$ towards 0 and 1.

It is important to recognize that attempting to induce sparsity in the solution to (8) is prone to introducing non-convexity to the program. As mentioned previously, integer programming with a branch and bound algorithm is non-convex. The ℓ_1-norm, when added as a linear regularization penalty to an objective function, is known to incentive sparsity in the solution [11][12]. However, due to the linear constraints in (8), ℓ_1 regularization is ineffective (i.e. $\|\hat{w}\|_1 = 1$). Direct attempts to drive the terms in $\hat{w}$ towards 0 and 1 (i.e. min $F(\mathbf{Y}^T\hat{w}) + \sum \hat{w}_j(1-\hat{w}_j)$) or to minimize the variance of $\tilde{y}$ (i.e. min $F(\mathbf{Y}^T\hat{w}) + \sum \hat{w}_j(y_j - \mathbf{Y}^T\hat{w})^2$) are concave.

In the remainder of this section, we present an iterative technique for inducing sparsity in $\hat{w}^*$. Put simply, at each iteration n, we solve (8) with a linear weight $\beta^n \in \mathbf{R}^{N_d}$ applied to $\hat{w}^n$

$$\begin{aligned} \underset{\hat{w}^n}{\text{minimize}} \quad & F(\mathbf{Y}^T\hat{w}^n) + \alpha(\hat{w}^n)^T\beta^n \\ \text{subject to} \quad & \textstyle\sum \hat{w}_j^n = 1 \\ & \hat{w}^n \geq 0 \\ & \hat{w}^n \in \mathbf{R}^{N_d} \end{aligned} \tag{20}$$

where α is a scaling parameter for the sparsity-inducing cost.

The linear weight is initialized at 0 ($\beta^0 \in \{0\}^{N_d}$) and after each iteration n, updated according to the previous solution $(\hat{w}^n)^*$

$$\begin{aligned} \beta_j^{n+1} = \frac{1}{N_d} - (\hat{w}_j^n)^* \\ \forall\, j = 1, \ldots, N_d \end{aligned} \tag{21}$$

For each successive iteration, β^n drives the terms in $\hat{w}^n$ away from $1/N_d$. If $(\hat{w}_j^n)^* > 1/N_d$, then $\beta_j^{n+1} < 0$ and the program will be incentivized to increase $\hat{w}_j^{n+1}$ in the next iteration. Inversely, if $(\hat{w}_j^n)^* < 1/N_d$ and $\beta_j^{n+1} > 0$, the program will try to decrease $\hat{w}_j^{n+1}$. Thus, the terms in $\hat{w}^n$ are encouraged, though not required, to approach 0 or 1. To enable tie-breaking, we can add a small random perturbation to the weight update

$$\begin{aligned} \beta_j^{n+1} = \frac{1}{N_d} - (\hat{w}_j^n)^* + v \\ \forall\, j = 1, \ldots, N_d \end{aligned} \tag{22}$$

where $v \in \mathbf{R}$ is a Gaussian random variable with a small covariance (e.g. $v \sim N(0, 0.01)$). This will allow ties between different output trajectories to be broken randomly.

This iterative technique for inducing sparsity $\hat{w}^*$ can be applied to the optimization of individual agents. While the objective of (20) is not constant, the change in β^n from one iteration to the next is relatively small. On the whole, the updating weight β^n introduces concavity into the problem. Specifically, the magnitude of each weight increases as the terms in $(\hat{w}^n)^*$ approach 0 or 1. With each successive iteration, $(\hat{w}^n)^*$ is forced further away from $(\hat{w}^0)^*$, the optimal solution to (8).

6. Illustrative Example

To illustrate the application of the ACT representation for the convex optimization of a non-convex discrete energy system, this section considers the control of a thermostatically controlled load (TCL). Specifically, we optimize the electricity demand of a simulated residential refrigerator using the techniques described in this manuscript.

The TCL is modeled using the hybrid state discrete time model [10][13][14]

$$\begin{aligned} T^k &= \theta_1 T^{k-1} + (1-\theta_1)(T_a^k + \theta_2 m^k) + \theta_3 \\ m^k &= \begin{cases} 1 & \text{if } T^k < T_{set} - \frac{\delta}{2} + u^k \\ 0 & \text{if } T^k > T_{set} + \frac{\delta}{2} + u^k \\ m^k & \text{otherwise} \end{cases} \end{aligned} \tag{23}$$

where state variables $T^k \in \mathbf{R}$ and $m^k \in \{0, 1\}$ denote the temperature of the conditioned mass and the discrete state (on or off) of the mechanical system, respectively. Additionally, $k = 1, 2, \ldots, N_t$ denotes the integer-valued time step, $T_a^k \in \mathbf{R}$ the ambient temperature (°C), $T_{set} \in \mathbf{R}$ the temperature setpoint (°C), and $\delta \in \mathbf{R}$ the temperature deadband width (°C). The control input $u^k \in \mathbf{S}_u$ is a setpoint change at each time step where $\mathbf{S}_u$ defines the discrete set of feasible values.

The parameter θ_1 represents the thermal characteristics of the conditioned mass as defined by

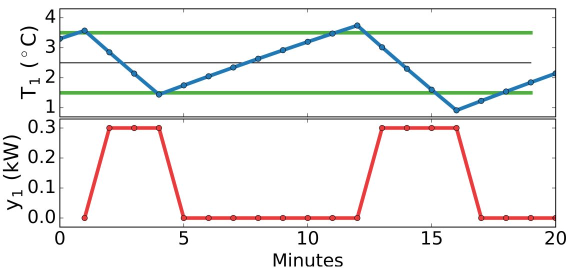

Figure 1: T_1 and y_1 trajectories given u_1

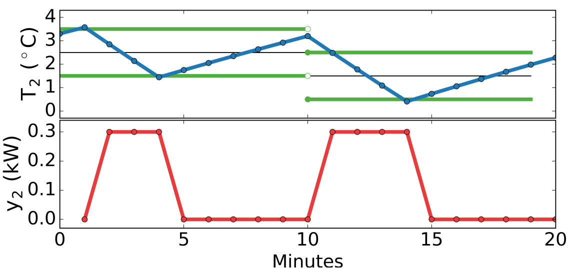

Figure 2: T_2 and y_2 trajectories given u_2

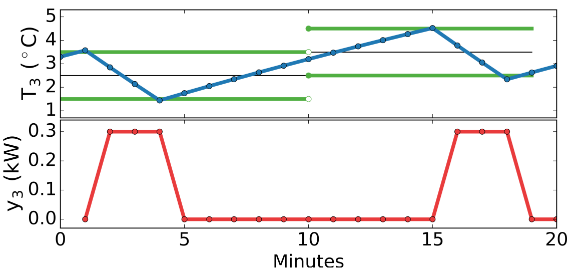

Figure 3: T_3 and y_3 trajectories given u_3

$\theta_1 = \exp(-h/RC)$ where C is the thermal capacitance (kWh/°C) and R is the thermal resistance (°C/kW), θ_2 the energy transfer to or from the mass due to the systems operation as defined by $\theta_2 = RP$ where P is the rate of energy transfer (kW), and θ_3 is an additive process noise accounting for energy gain or loss not directly modeled.

The electricity demand of the TCL at each time step is defined by

$$y^k = \frac{|P|}{COP} m^k \tag{24}$$

where $y^k \in \mathbf{R}$ is the electric power demand (kW) and COP the coefficient of performance. We now have the state and output equations necessary to model the system ((23) serves as G and (24) as H).

Figures 1, 2, and 3 present examples of $N_a = 3$ alternative trajectories for the TCL. In the examples, each alternative input u_j for $j = 1, 2, 3$ is $\in \{0, -1, 1\}^{20}$ (i.e $N_t = 20$). While the input trajectories are not plotted, they can be inferred from the changes in the setpoint and temperature bounds. For trajectory $j = 1$, $u_1^k = 0$ for $k = 1, \ldots, 20$. For trajectory $j = 2$, $u_2^k = 0$ for $k = 1, \ldots, 10$ and $u_2^k = -1$ for $k = 11, \ldots, 20$. For trajectory $j = 3$, $u_3^k = 0$ for $k = 1, \ldots, 10$ and $u_3^k = 1$ for $k = 11, \ldots, 20$.

The TCL has been simulated using (23) and (24) with a default setpoint T_{set} of 2.5°C, a deadband width δ of 2°C, an initial temperature T^0 of 3.3°C, and an initial mechanical state m^0 of 0. Figures 1, 2, and 3 present the T_j and y_j trajectories corresponding to each input u_j for $j = 1, 2, 3$. The mechanical state trajectories m_j can be inferred from the T_j and y_j trajectories. As illustrated by the figures, each distinct input u_j produces a distinct T_j, m_j, and y_j. Therefore, in this example, $N_d = N_a = 3$.

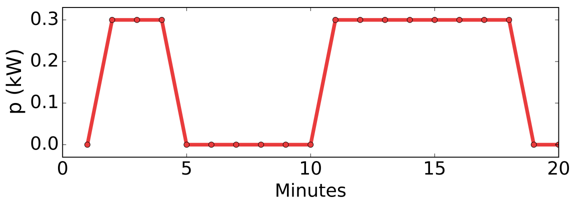

Figure 4: Target power demand p^*

Next, we define some optimal power demand trajectory $p \in \mathbf{R}^{20}$ which we would like the TCL to match as closely as possible. As illustrated in Figure 4, we define $p^k = 0.3$ for $k = 2, \ldots, 4$ and for $k = 11, \ldots, 18$ and $p^k = 0$ otherwise. The convex optimization program is defined with a least squares objective function

$$\begin{array}{ll} \underset{\hat{w}}{\text{minimize}} & \|\mathbf{Y}^T \hat{w} - p^*\|_2^2 \\ \text{subject to} & \sum \hat{w}_j = 1 \\ & \hat{w} \geq 0 \\ & \hat{w} \in \mathbf{R}^{N_d} \end{array} \tag{25}$$

By solving (25) with Y and p as described above, we find that $\hat{w}^* = (0.263, 0.421, 0.316)$. The continuous solution $\hat{y}^*$, the optimal linear combination of the alternative output trajectories, is illustrated in Figure 5.

It should be noted that the squared error between p and y_1, y_2, and y_3 is 0.134, 0.134, and 0.15, respectively. Thus, the utility of y_1 and y_2 are equal.

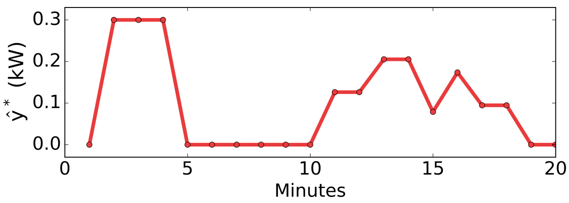

Figure 5: Continuous solution $\hat{y}^*$

However, if we apply (9), there are 3 possible outcomes for the discrete solution $\tilde{w}$,

$$\begin{aligned} \Pr(\tilde{w} = (1,0,0)) &= 26.3\% \\ \Pr(\tilde{w} = (0,1,0)) &= 42.1\% \\ \Pr(\tilde{w} = (0,0,1)) &= 31.6\% \end{aligned} \tag{26}$$

By applying the sparsity inducing penalty described in (20) and (22), we find that the $(\hat{w}^*)^n = (0,1,0)$ after 3 or 4 iterations. Despite the random perturbation added to the weights, we observe that, for this particular example, the program always converges to the same solution (i.e $(\hat{w}^*)^n \to (0,1,0)$ as $n \to \infty$). Thus, for this TCL, we would implement the control trajectory defined by u_2, T_2, m_2, and y_2.

7. Conclusions

In this manuscript, we have developed the alternative control trajectory (ACT) representation – a novel approach for representing the control of a non-convex discrete system as a convex program. The resulting convex program provides a solution that can be interpreted stochastically for implementation. This approach enables the approximate optimal control of non-convex agents using distributed convex optimization techniques. By inducing sparsity in the individual agents, we can increase the predictability (i.e. reduce the variance) of the aggregated output.

8. References

[1] Y. V. Makarov, C. Loutan, J. Ma, P. De Mello, Operational impacts of wind generation on california power systems, Power Systems, IEEE Transactions on 24 (2) (2009) 1039–1050.

[2] Z. Xu, J. Ostergaard, M. Togeby, C. Marcus-Moller, Design and modelling of thermostatically controlled loads as frequency controlled reserve, in: Power Engineering Society General Meeting, 2007. IEEE, IEEE, 2007, pp. 1–6.

[3] B. J. Kirby, Frequency regulation basics and trends, ORNL/TM 2004/291, Oak Ridge National Laboratory.

[4] E. Hirst, B. Kirby, Separating and measuring the regulation and load-following ancillary services, Utilities Policy 8 (1999) 75–81.

[5] G. Strbac, Demand side management: Benefits and challenges, Energy Policy 36 (12) (2008) 4419–4426.

[6] D. S. Callaway, I. Hiskens, et al., Achieving controllability of electric loads, Proceedings of the IEEE 99 (1) (2011) 184–199.

[7] C. Le Floch, F. di Meglio, S. Moura, Optimal charging of vehicle-to-grid fleets via pde aggregation techniques, Power 1000 1100.

[8] E. M. Burger, S. J. Moura, Generation Following with Thermostatically Controlled Loads via Alternating Direction Method of Multipliers Sharing Algorithm (October 2015).
URL `http://escholarship.org/uc/item/2m5333xx`

[9] J. L. Mathieu, M. Dyson, D. S. Callaway, Using residential electric loads for fast demand response: The potential resource and revenues, the costs, and policy recommendations, ACEEE Summer Study on Energy Efficiency in Buildings.

[10] D. S. Callaway, Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy, Energy Conversion and Management 50 (5) (2009) 1389–1400.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[12] E. J. Cands, M. Wakin, S. Boyd, Enhancing sparsity by reweighted l1 minimization, Journal of Fourier Analysis and Applications 14 (5) (2008) 877–905.

[13] S. Ihara, F. C. Schweppe, Physically based modeling of cold load pickup, IEEE Transactions on Power Apparatus and Systems 100 (9) (1981) 4142–4250.

[14] R. E. Mortensen, K. P. Haggerty, A stochastic computer model for heating and cooling loads., IEEE Transactions on Power Systems 3 (3) (1998) 1213–1219.