Benchmarks for Grid Energy Management with Python Gekko

Nathaniel S. Gates[1], Daniel C. Hill[1], Blake W. Billings[2], Kody M. Powell[2], and John D. Hedengren[1]

***Abstract*— Recent grid energy problems in California and Texas highlight the need for optimized grid design and operation. The complexity of the electric grid presents difficult control problems that require powerful solvers and efficient formulations for tractable solutions. The Gekko Optimization Suite is a machine learning and optimization package in Python for optimal control with differential algebraic equations and is capable of solving complex grid design and control problems. A series of non-dimensional benchmark cases are proposed for grid energy production. These include (I) load following, (II) cogeneration, (III) tri-generation, and energy storage with (IV) constant production, (V) load following, and (VI) cogeneration. Individual case studies include ramp rate constraints, power production, and energy storage operation as design variables. The tutorials demonstrate methods to solve control problems with sequential and simultaneous solutions of the objective and dynamic constraints. While these tutorials are specific to grid energy system optimization, the tutorials also demonstrate how to efficiently solve large-scale nonlinear dynamic systems with a trade-off analysis between sequential and simultaneous methods.**

I. Background

A. Literature Review

Modern society depends heavily on access to vast quantities of electrical energy. In 2020, the U.S. consumed more than 4 TWh of electricity [1]. There is strong interest in providing that energy in affordable, reliable, clean and sustainable ways. As a result of cost declines and policy support, power generation from renewable energy sources (such as wind and solar) has increased dramatically in recent years [2]. These intermittent renewable energy sources can pose increased challenges to the stability of the power grid [3].

Modern solutions to operating power grids with a growing share of renewable energy generation require optimal control of dispatchable generators to respond to increased uncertainty and continuously provide stable and reliable electricity. Because of this, flexibility is increasingly becoming a valued characteristic for generators, consumers, and storage applications. Due to the complexity of optimal grid energy management, it is useful to consider benchmark problems for optimal control of dispatchable generators in the power grid. However, many existing benchmarks and models in the energy modeling literature lack flexibility and are too specific to certain applications and systems to be generally applicable [4].

Energy storage systems are a potential solution to the problems imposed by intermittent renewable energy sources. Cost declines in lithium-ion batteries will likely lead to more widespread adoption of these storage methods.According to the U.S. Department of Energy, energy storage markets are estimated to grow to three-to-five times the current size by 2030 [5]. Optimizing these storage systems (including Li-ion battery storage, pumped-storage hydropower, thermal energy storage, and others) requires charging and dispatch optimization to fulfill financial and energy benefit metrics. Classic control and optimization for energy storage utilizes Kalman filters and the Linear Quadratic Regulator for state estimation and to optimize operation [6]. Other more robust modeling suites like REopt from the National Renewable Energy Laboratory (NREL) utilize large-scale solvers such as Xpress, CBC, and CPLEX to solve mixed-integer linear programs (MILP) associated with energy storage [7].

Benchmarks are used in optimization to show solver capabilities without the difficulties introduced with high problem complexity. Multi-objective optimization problems have long been a challenge to solve but can be simplified by scalable and customizable benchmarks problems [8]. Benchmark problems are a simple standard to judge capabilities of solvers and algorithms to optimize real world problems. For example, the IEEE Congress on Evolutionary Computation (CEC) regularly holds competition on large-scale global optimization where participants test their optimization algorithms on fifteen established benchmark functions [9]. In energy modeling, many studies create benchmarks to simplify complex systems such as whole-building energy simulation or sector-coupled energy systems [4], [10].

The focus of this work is to propose benchmark problems for grid energy management and demonstrate solutions with the Gekko Optimization Suite [11], [12]. These benchmark problems for optimal control of dispatchable generators are simple by design to show the progressive capabilities of Gekko in modeling energy systems. More complex grid energy management case studies are available to demonstrate performance on realistic design and dispatch optimization scenarios [13], [14], [15], [16].

B. Python Gekko

Gekko [12] is a Python package which interfaces with the APMonitor modeling language [17] and is capable of solving large-scale dynamic optimization problems with mixed-integer and differential algebraic equations. It is open-source and provides easy access to solvers such as IPOPT [18], and can also be used with commercial solvers. Gekko through APMonitor performs automatic differentiation to provide

[1]Department of Chemical Engineering, 330 EB Brigham Young University, Provo, Utah

[2]Department of Chemical Engineering, 50 S. Central Campus Dr., University of Utah, Salt Lake City, Utah

exact gradients and has several different solve modes for both steady-state and dynamic simulation, estimation, and control [12]. It discretizes the model equations in time according to the provided time horizon, and uses direct transcription to convert the differential equations into a system of algebraic equations, using orthogonal collocation on finite elements the solution between the requested time points in the horizon. This allows efficient large-scale nonlinear optimizer to solve dynamic optimization problems, which are notoriously difficult to solve due to the high-dimensionality caused by the time-dependence [19].

C. Dynamic Solve Modes

For dynamic optimization problems, Gekko has simultaneous and sequential solve modes [19]. In the simultaneous mode, the solver attempts to converge both the objective function and the model equations together. The simultaneous solution is enabled by orthogonal collocation on finite elements to convert the differential equations into algebraic expressions. It iterates until convergence within the specified tolerance using the chosen norm, typically either the 1-norm or 2-norm. Because the solver is able to optimize the entire time horizon, it can respond to complicated changes in the dynamics and predict the quadratic effect of variable changes on time points into the future with gradient and Hessian information. This approach is thus well equipped to handle problems with a large number of degrees of freedom (decision variables in excess of the number of equations). Once the problem converges, the solution is a local optimal solution, but until then the solution is meaningless (i.e. infeasible).

The second solution method is the sequential solve mode (shooting method). In this method, the objective function and the model equations are solved separately in a sequential fashion. In each major iteration, the parameter estimates (degrees of freedom) are held constant, and the equations are evaluated in a simulation mode. A series of minor iterations then take place, in which the objective function is evaluated and exact first and second derivatives of the objective function with respect to the parameter estimates are computed. These are used to find a new search direction and obtain new parameter estimates. This minor iteration process repeats as the solver seeks to converge the objective function to within the specified tolerance using the chosen norm. If it is unsuccessful, a new major iteration occurs, and the process continues. The sequential solve mode typically does well in problems with a large number of states (i.e. model equations) and a low number of degrees of freedom. One of the benefits of this mode is that if the optimization process is terminated early, the solution is still feasible, though sub-optimal.

II. Benchmark Models

The benchmark problems have intuitive solutions and serve to illustrate the optimization principles. They provide insight that can be leveraged to solve more challenging dynamic optimization problems when the solution is not intuitive and the process to achieve a reliable converged solution is lengthy and difficult. They also provide a convenient way to exhibit and compare the capabilities of different solution methods and algorithms.

The benchmark problems considered here have been chosen to represent unique features of grid energy systems. They are convex, non-dimensionalized for simplicity, and have features that are common to other grid energy systems. They capture trade-offs in coordinating dispatch and control, including perfect forecasting, ramp-rate-constrained decision making, generating multiple products, and energy storage. While the solutions are relatively straightforward and the forecasts have no uncertainty, the problems provide a framework for understanding the varying characteristics of grid energy management problems and investigating trade-offs between different solution methods. The first three benchmarks deal with ramp constraints, and the last three with energy storage. Symbols used in the benchmark problems are defined in Table I.

TABLE I
Symbols used in the benchmark cases

Symbol	Description
J	objective function
d	demand
g	generation
r	ramp rate
e	storage inventory
q_{in}, q_{out}	energy stored, recovered
R	renewable source
s_{out}, s_{in}	slack variables for storage switching
η	storage efficiency
n	number of generating units
$(\ )_i$	subscript indicates product i

A. Benchmark I: Load Following

The first benchmark problem represents load following, a common scenario in grid systems. The optimizer seeks to match demand and supply with fluctuating demand dynamics. A single generator with ramping constraints attempts to respond to a single load. In this and all other problem formulations, it is assumed that future demand is perfectly known. The generation and demand match initially, but the generator must ramp in order to ensure this throughout the horizon while minimizing overproduction. The formulation is shown in Eq. 1:

$$\min_r J = \int_{t=0}^{1} [1000 \max(0, d - g) + \max(0, g - d)]\, dt \tag{1a}$$

$$\text{s.t.} \quad \frac{dg}{dt} = r \tag{1b}$$

$$d = \cos(2\pi t) + 3 \tag{1c}$$

$$-1 \le r \le 1 \tag{1d}$$

$$g(0) = 4 \quad (d(0) = 4) \tag{1e}$$

As noted previously the values are dimensionless for simplicity and generality. The electricity demand for the system is a sinusoidal fall and rise (1c). The first term in the objective function, $\max(0; d - g)$, represents the under-production of electricity (i.e. dropped load) and is given a cost of 1,000. The second term in the objective function, $\max(0; g - d)$, represents the overproduction of electricity and is given a cost of 1. This difference in cost reflects the difference in severity between underproduction and overproduction of electricity. As is typical for steam-cycle power generators, there is a ramp rate r that constrains load following, which in this case limits the generation rate of change to between -1 and 1 (1d). The specific values in this case study have no direct relation to an actual physical system, but they highlight a common type of objective function and constraint associated with grid dispatch optimization. This is the simplest benchmark case with a single producer and electrical-only demand.

The optimal solution to Benchmark I is shown in Fig. 1. The generator immediately ramps down at its maximum rate, seeking to minimize overproduction until $t = 0.5$, when it ramps back up to meet the demand at $t = 1$. There is overproduction throughout the horizon except at the very beginning and ending, when the optimizer is able to meet the demand exactly. Even though overproduction is the highest at $t = 0.5$, the optimizer increases the generation then in order to meet the future demand constraint at $t = 1$ and avoid any dropped load. This amount of overproduction would very likely not occur in a physical system, but this problem serves to illustrate the dynamics of load following.

B. Benchmark II: Cogeneration

In the second benchmark problem, one producer seeks to meet two objectives that are constraining at different times. Benchmark II enhances Benchmark I by replacing the generator with a cogeneration system ($n = 2$) that produces (1) electricity and (2) heat in response to electricity demand and a new heat demand profile. Both products are included in the objective function summation as shown in Eq. 2.

$$\min_r J = \sum_{i=1} \int_{t=0}^{1} 1000 \max(0; d_i - g_i) \tag{2a}$$
$$+ \max(0; g_i - d_i) \, dt$$

$$\text{s.t.} \quad \frac{dg_1}{dt} = r; \quad g_2 = 2g_1 \tag{2b}$$
$$d_1 = \cos(2\pi t) + 3 \tag{2c}$$
$$d_2 = 1.5 \sin(2\pi t) + 7 \tag{2d}$$
$$-1 \leq r \leq 1 \tag{2e}$$
$$g_1(0) = 4 \quad (d_1(0) = 4) \tag{2f}$$
$$g_2(0) = 8 \quad (d_2(0) = 7) \tag{2g}$$

Here, (1) and (2) are generated simultaneously, with twice as much (2) generated as (1) (Eq. 2b). The same over- and under-generation penalties as before are applied now to both products. The heat (2) demand profile (Eq. 2d) is offset by a quarter cycle from the electricity (1) demand profile (Eq. 2c) to represent the situation where both electric power and heat alternate as the driving force for load following.

The optimal solution to Benchmark II is shown in Fig. 2. At $t = 0$, the electricity demand is met exactly and the waste heat exceeds the requirement. Although the electricity demand then decreases, the increasing heat demand in the future drives the system to increase production at its maximum rate. At $t = 0.25$ the production decreases, responding to the lower heat demand, but at $t = 0.75$ production ramps up again in order to meet the increasing electricity demand at $t = 1$. As before, the optimal solution avoids any underproduction, though with an increased amount of overproduction. The ramp rate constraint (2e) plays a key role in determining the system dynamics, driving not only how well the system meets the overall objective but which part of the objective is the driving force at which times.

C. Benchmark III: Tri-generation

The third benchmark problem enhances the previous problems further still, creating a tri-generation system ($n = 3$) with two producers, three products, and three demand profiles. The primary producer is the same as the prior benchmark and is ramp-rate constrained to produce the two primary products (e.g., electricity and heat). An additional producer (e.g., a solid oxide electrolysis cell) uses these

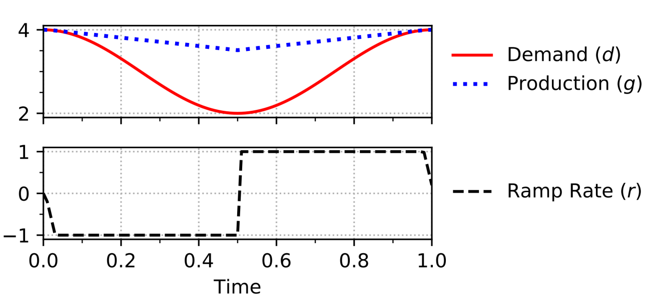

Fig. 1. Optimal solution to *Benchmark I: Load Following* with ramping constraints, a single producer and a demand profile. The example product is electricity.

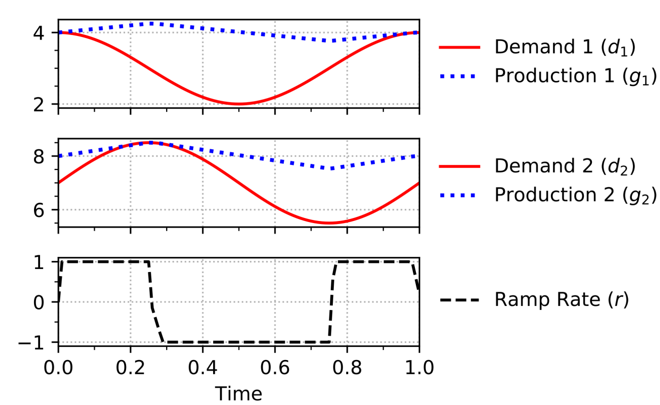

Fig. 2. Optimal solution to *Benchmark II: Cogeneration* with ramping constraints, two demand profiles, and one cogeneration producer of products 1 and 2. The example products are electricity (1) and heat (2).

first two products to make a third product (e.g., hydrogen), thereby utilizing any excess system capacity and maximizing its production while avoiding supply shortages for products one and two. A mathematical statement of the third benchmark problem is given by Eq. 3:

$$\min_{r_1, r_3} \; J = \sum_{i=1} \int_{t=0}^{1} 1000 \; \max(0, d_i - g_i) + \max(0, g_i - d_i) \; dt - 0.1 \int_{t=0}^{1} g_3 \, dt \tag{3a}$$

$$\text{s.t.} \quad \frac{dg_1}{dt} = r_1, \; \frac{dg_3}{dt} = r_3 \tag{3b}$$

$$g_2 = 2 g_1 \tag{3c}$$

$$d_1 = \cos(2\pi t) + 3 + 2 g_3 \tag{3d}$$

$$d_2 = 1.5 \sin(2\pi t) + 7 + 3 g_3 \tag{3e}$$

$$d_3 = \max[0, -0.2 \sin(2\pi t)] \tag{3f}$$

$$-1 \le r_1 \le 1, \quad -1 \le r_3 \le 1 \tag{3g}$$

$$g_1(0) = 4 \quad (d_1(0) = 4) \tag{3h}$$

$$g_2(0) = 8 \quad (d_2(0) = 7) \tag{3i}$$

$$g_3(0) = 0 \quad (d_3(0) = 0) \tag{3j}$$

The three example products are (1) electricity, (2) heat and (3) hydrogen. The first producer (e.g., cogeneration system) produces (1) and (2) and the second producer (e.g., a solid oxide electrolysis cell) produces (3). These producers are coupled because the products of the first are the inputs of the second. Additionally, they both have the same ramp rate constraint (3g). Producing one unit of (3) requires two units of (1) and three units of (2) (3c). The demand for (3) is given in (3f). Any excess capacity of (1) and (2) are used to produce additional (3) (3d, 3e). As before, over- and under-generation penalties are applied now to all three products (3a). A term is also added to the objective function to incentivize hydrogen (3) production without causing overproduction of (1) and (2).

The optimal solution to Benchmark III is shown in Fig. 3. For the first producer, the dynamics from $t = 0$ to $t = 0.5$ are similar to Benchmark II, with load following of first (1) and then (2) demand driving the optimal solution. In the second half of the horizon, the demand for (3) causes increased production beyond the needs of the final demand for (1), leading to an overproduction of (1) and then a decrease in production at $t = 0.95$ down to the final constraint at $t = 1$.

For the second producer, the excess supply of (1) and (2) from $t = 0$ to $t = 0.2$ caused by the ramp-constrained load following of producer one is used to produce (3), with producer two ramping up and down at its maximum ramp rate. Here, the constraints on the demand for (2) limit the production of (3). Once again, the actual values here simply serve to illustrate the dynamics, as this much overproduction would not occur in a typical system.

Producer two consumes more (2) than (1), and thus it isn't until the peak demand for (2) at $t = 0.3$ is reached that producer two can increase the production of (3), enabling it to use more of the excess of (1) once sufficient excess of

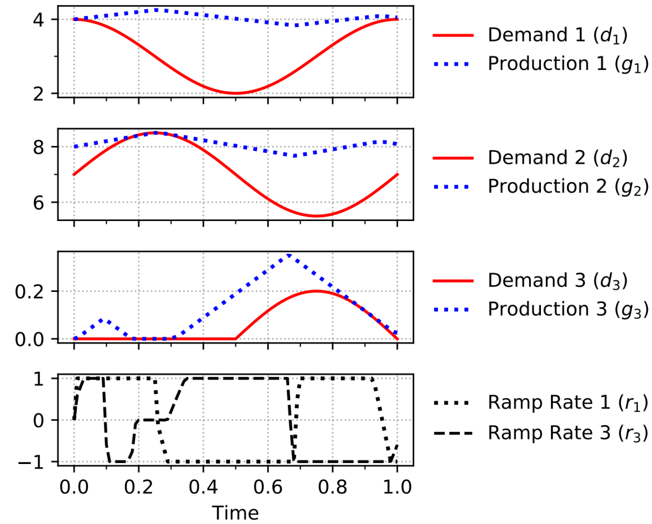

Fig. 3. Optimal solution to *Benchmark III: Tri-generation* with ramping constraints, three demand profiles, and two producers generating products 1-3. The example products are electricity (1), heat (2) and hydrogen (3).

(2) is generated. It does so at its maximum ramp rate, and produces (3) all the way until $t = 0.65$, where it decreases production at its maximum ramp rate until nearly the end of the time horizon. By so doing, producer two is able to exactly meet the tail end of the hydrogen demand at $t = 0.9$.

Again, this problem is not specific to any particular system, but the elements are similar to those found in dispatch optimization problems such as co-generation of electricity and heat combined with chemical production. In addition, many district-wide systems produce electricity, heat, and cooling and are another example of tri-generation. In both cases, the third product adds demand for the other products.

D. Benchmark IV: Constant Production with Energy Storage

The fourth benchmark problem models a hybrid system with a single generator with constant production constraints coupled with energy storage that together must meet an oscillating electricity demand. The goal of the problem is to minimize the required power production and use energy storage to capture excess generation serve the oscillating energy demand while keeping the generator production constant. In order to prevent the energy storage from charging and discharging simultaneously without requiring mixed-integer variables, slack variables are used to control when the storage charges and discharges, allowing it to switch modes in a way that is both continuous and differentiable. This allows the modeling language to use automatic differentiation to provide exact gradients to the solver. The formulation of Benchmark IV is shown in Eq. 4, and is adapted from [16].

$$\min_{g} \; g \tag{4a}$$

$$\text{s.t.} \quad \frac{de}{dt} = q_{\text{in}} - q_{\text{out}} \tag{4b}$$

$$q_{\text{in}} = g - d + s_{\text{in}} \tag{4c}$$

$$q_{out} = d \quad g + s_{out} \tag{4d}$$
$$g \quad d = s_{out} \quad s_{in} \tag{4e}$$
$$s_{out}; s_{in} \quad 0; \; q_{out} \quad q_{in} \quad 0 \tag{4f}$$
$$g + q_{out} = \quad q_{in} \quad d \tag{4g}$$
$$e \quad 0; \quad = 0{:}7 \tag{4h}$$
$$d = 10 \quad 2\sin(2 \quad t) \tag{4i}$$
$$e(0) = e(1) = 0 \tag{4j}$$

The optimal solution to Benchmark IV is shown in Fig. 4. The production is higher than the initial demand because the solver accounts for storage efficiency losses (4h) throughout the horizon. A periodic constraint ensures that the storage starts and ends empty (4j). The solver uses all of the stored capacity to most effectively minimize the production level. Though this problem is not specific to a particular system, generators that have difficulty ramping (such as nuclear plants) could use energy storage to meet a changing demand profile while retaining constant production.

E. Benchmark V: Load Following with Energy Storage

The fifth benchmark combines energy storage with a load-following problem similar to Benchmark I. The first-half of the time horizon is nearly identical to Benchmark I, but now the excess energy can be stored. This allows the system to meet a higher demand in the second half of the time horizon without needing extremes in generation. The solver minimizes the ramping needs and operates more flexibly by storing and then recovering the overproduction caused by the ramping constraints. Energy storage allows this generator to meet the load without requiring significant overproduction.

$$\min_{r} \; J = \sum_{i=1} \int_{t=0}^{1} [1000 \max(0; \quad) + \max(0; \quad)]dt$$
$$\text{where} \quad = d \quad g \quad R + q_{out} = \quad q_{in} \tag{5a}$$
$$s{:}t{:} \quad \frac{de}{dt} = q_{in} \quad q_{out} \tag{5b}$$
$$q_{in} = g + R \quad d + s_{in} \tag{5c}$$
$$q_{out} = d \quad g \quad R + s_{out} \tag{5d}$$
$$g + R \quad d = s_{out} \quad s_{in} \tag{5e}$$
$$s_{out}; s_{in} \quad 0; \; q_{out} \quad q_{in} \quad 0 \tag{5f}$$
$$g + R + q_{out} = \quad q_{in} \quad d \tag{5g}$$
$$e \quad 0; \quad = 0{:}85 \tag{5h}$$
$$d = 7 \quad 2\sin(2 \quad t) \tag{5i}$$
$$R = \begin{cases} 3 + 3\cos(4 \quad t) & \frac{1}{4} \quad t \quad \frac{3}{4} \\ 0 & \text{otherwise} \end{cases} \tag{5j}$$
$$\frac{dg}{dt} = r; \quad 1 \quad r \quad 1 \tag{5k}$$
$$e(0) = e(1) = 0 \tag{5l}$$

The formulation is a combination of Benchmarks I and IV. The objective function is similar to that of Benchmark I with added terms for renewable generation and storage (5a). Renewable generation is added as a piecewise function of time (5j). The optimal solution to Benchmark V is shown in Fig. 5.

The addition of a renewable source drastically changes the demand profile. The implementation of storage allows the producer to ramp less often while meeting a more extreme demand profile. This is evident in the ramp rate remaining constant through the first half of the time horizon and only changing once to meet the steep demand increase as the renewable source drops. Only a moderate production increase is needed because the storage dispatch eases the ramping needs. The solver efficiently uses the storage to meet the net demand and reduces the changes in ramping. This benchmark is similar to storage mediating the high evening ramp needs of the "duck curve" created by solar photovoltaic (PV) generation during the day.

F. Benchmark VI: Cogeneration with Dual Energy Storage

The sixth benchmark problem is a combination of Benchmarks II and V where the ramp rate of the generator is the manipulated variable but now must meet both electrical (1) and heat (2) demand with the use of both electrical and thermal storage. The formulation of Benchmark VI is shown in Eq. 6. A renewable generation source (6k) (such as solar PV) is added to the system as an auxiliary electrical energy source that cannot be controlled. The objective is the same as in Benchmark IV, to minimize power production (6a), but this time while meeting both the heat and power demands (6i, 6j).

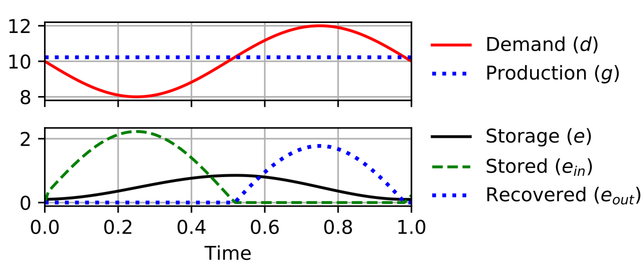

Fig. 4. Optimal solution to *Benchmark IV: Constant Production with Energy Storage* with a constant production producer, a demand profile, and storage. The example product is electricity.

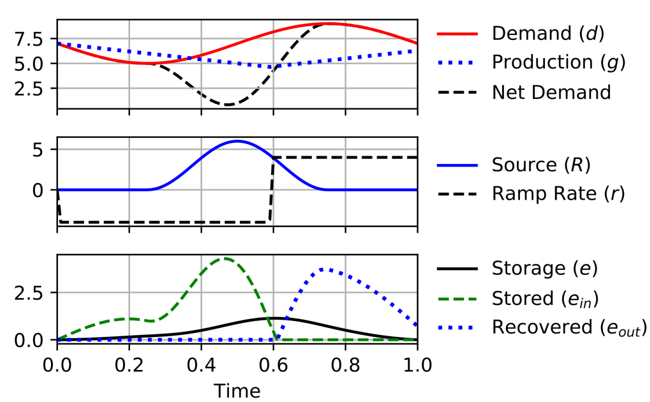

Fig. 5. Optimal solution to *Benchmark V: Load following with Energy Storage* with ramping constraints, a producer, a demand profile, a renewable product source, and product storage. The example product is electricity.

Slack variables are used in the same way as in Benchmarks IV and V.

$$\min_{r_1} \ g_1 \tag{6a}$$
$$\text{s.t.} \ \frac{de_i}{dt} = e_{\text{in},i} \eta_i - e_{\text{out},i} \tag{6b}$$
$$e_{\text{in},i} = g_i + R_i - d_i + s_{\text{in},i} \tag{6c}$$
$$e_{\text{out},i} = d_i - g_i - R_i + s_{\text{out},i} \tag{6d}$$
$$g_i + R_i - d_i = s_{\text{out},i} - s_{\text{in},i} \tag{6e}$$
$$s_{\text{out},i}, s_{\text{in},i} \geq 0, \ e_{\text{out},i} \cdot e_{\text{in},i} \leq 0 \tag{6f}$$
$$g_i + R_i + e_{\text{out},i} = \eta_i e_{\text{in},i} \geq d_i \tag{6g}$$
$$e_i \geq 0, \ \eta_1 = 0.7, \ \eta_2 = 0.8 \tag{6h}$$
$$d_1 = 10 - 2\sin(2\pi t) \tag{6i}$$
$$d_2 = 15 + 1.5\cos(2\pi t) \tag{6j}$$
$$R_1 = \begin{cases} 2 + 2\cos(4\pi t) & \frac{1}{4} \leq t \leq \frac{3}{4} \\ 0 & \text{otherwise} \end{cases} \tag{6k}$$
$$\frac{dg_1}{dt} = r_1, \ -3 \leq r_1 \leq 3 \tag{6l}$$
$$R_2 = 0, \ g_2 = 1.5 \, g_1 \tag{6m}$$
$$e_1(0) = 0, \ e_2(0) = e_2(1) = 0.5 \tag{6n}$$

The optimal solution to Benchmark VI is shown in Fig. 6. Initially, the electricity (1) production matches the demand, and the plant ramps up to more closely match heat demand, saving some of the stored heat (2) for $t = 0.5$ when the renewable source provides abundant power. The plant then ramps back down with perfect knowledge of the renewable source peak. At $t = 0.5$ the renewable source is high and the heat (2) demand is low, and a large portion of electricity (1) charging takes place. The generator then ramps up at its maximum level to meet the high electricity (1) demand while also using some of the stored electricity. This increased production is required to generate excess heat (2) for storage, providing enough heat storage to meet the final heat demand while satisfying the periodic heat storage constraint (6n). The optimal result shows the solver's ability to anticipate future demand of both products and utilize both storage capacities to minimize the overall electricity (1) production.

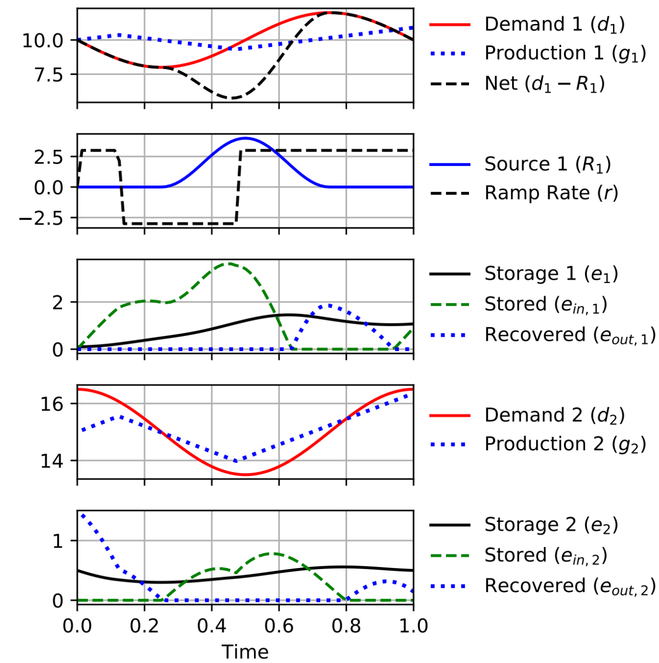

Fig. 6. Optimal solution to *Benchmark VI: Cogeneration with Dual Energy Storage* with ramping constraints, one cogeneration producer of two products, two demand profiles, a renewable source of product 1, and storage for both products. The example products are electricity (1) and heat (2).

III. Dispatch Benchmark Case Summary and Conclusions

The first five benchmark problems are solved in a simultaneous mode using an ℓ_1-norm objective over a horizon of 101 time points. The sixth benchmark is solved using an ℓ_2-norm objective over a horizon of 73 time points. This benchmark produced slightly different results when solved with the ℓ_1-norm versus the ℓ_2-norm. To be consistent, the ℓ_1-norm is used for all of the benchmark problems in the grid refinement study.

To benchmark the performance of Python Gekko, all six benchmark problems are solved in the simultaneous solution mode across a time horizon of increasing resolution. The first three benchmark models are also solved in the sequential solve mode. They are solved with only two collocation nodes, which increased the solve speed without sacrificing too much accuracy in the solutions. These performance benchmarks are run on a Dell R815 Server with an AMD Opteron Processor 6276, 64 CPUs, 64 GB of RAM, and RAID array 15k RPM hard drives. The optimization results are shown in Fig. 7. The benchmark problems are of increasing difficulty with the exception of Benchmark IV, which has the fastest solve time. Although Benchmark IV includes energy storage and uses slack variables, the production decision variable is constant across the entire horizon instead of a dynamic decision variable.

In each case, the simultaneous solve mode performed better than the sequential solve mode, as expected. For the first two benchmarks, the sequential solve mode performed comparably to the simultaneous mode in the first few time horizons. This is because the problems are simple enough for the solver to converge in one major iteration. As the degrees of freedom increase with the time horizon resolution, the sequential solve method iterates more and more to converge the optimization problems, and the performance suffers. By contrast, the simultaneous mode scales well as the problem size grows, handling large numbers of degrees of freedom. For these types of problems, the model equations do not provide time savings in the sequential mode, and the number of degrees of freedom from the dynamic horizon have the greater complexity.

Benchmarks I-IV have degrees of freedom that increase with time resolution. Benchmarks V-VI have negative degrees of freedom if all slack variables are at the constraints.