One Robot for Many Tasks: Versatile Co-Design Through Stochastic Programming

Article *in* IEEE Robotics and Automation Letters · January 2020

DOI: 10.1109/LRA.2020.2969948

3 authors:

Gabriel Bravo Palacios
University of Notre Dame
5 PUBLICATIONS **99** CITATIONS
SEE PROFILE

Andrea Del Prete
University of Trento
78 PUBLICATIONS **2,317** CITATIONS
SEE PROFILE

Patrick Wensing
University of Notre Dame
147 PUBLICATIONS **5,893** CITATIONS
SEE PROFILE

One Robot for Many Tasks: Versatile Co-Design through Stochastic Programming

Gabriel Bravo-Palacios[1], *Student Member, IEEE*, Andrea Del Prete[2], *Member, IEEE*, and Patrick M. Wensing[1], *Member, IEEE*

***Abstract*—Versatility is one of the main factors driving the adoption of robots on the assembly line and in other applications. Compared to fixed-automation solutions, a single industrial robot can perform a wide range of tasks (e.g., welding, lifting). In other platforms, such as legged robots, versatility is a necessity to negotiate varied terrains. The ability to balance performance across these anticipated scenarios is one of the main challenges to the process of design. To address this challenge, this letter proposes a new framework for the computational design of versatile robots by considering the interplay between mechanical design and control across multiple tasks and environments. The proposed method optimizes morphology parameters while simultaneously adjusting control parameters using trajectory optimization (TO) so that a single design can fulfill multiple tasks. As its main contribution, the letter details an approach to combine methods from stochastic programming (SP) with TO to address the scalability of these multi-task co-design problems. To assess the effects of this contribution, this letter considers the problems of designing a planar manipulator to transport a range of loads and a hopping monopod robot that must jump across a variety of terrains. The proposed formulation achieves faster solution times and improved scalability in comparison to state of the art co-design solutions. The resulting designs are also shown to be more versatile in terms of providing improved energy cost and task completion times across multiple scenarios.**

***Index Terms*—Optimization and Optimal Control, Legged Robots, Mechanism Design.**

I. Introduction

A LONGSTANDING challenge in the field is designing robots to exhibit the same levels of mobility and dexterity as observed in animals. As a result of evolution, animals move through and manipulate the world via a synergy of body and brain, leveraging feedback pathways encoded in the structure of muscles, tendons, and ligaments to simplify neurocontrol. By contrast, the design of robot hardware (body) and associated control systems (brain) are often treated as disjoint problems. On one end, focus purely on design has resulted in systems such as passive dynamic walkers [1], which excel at one task at the expense of versatility. Achieving designs that excel in a range of scenarios requires consideration of how control can be actively used to enable adaptability.

Manuscript received: September, 10, 2019; accepted January, 6, 2020. This letter was recommended for publication by Associate Editor Y. Zhao and Editor D. Song upon evaluation of reviewers' comments.

[1]G. Bravo-Palacios and P. M. Wensing are with the Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN, 46556 USA. (e-mail: gbravopa@nd.edu, pwensing@nd.edu)

[2]A. Del Prete is with the Department of Industrial Engineering, University of Trento, Trento, 30123 Italy. (e-mail: andrea.delprete@unitn.it)

Digital Object Identifier 10.1109/LRA.2020.2969948

The interplay between morphology and control makes the design of robotic systems particularly challenging. Finding a design to fulfill even a single task can be a time consuming and largely manual process [2], [3]. Solutions to partially automate the design process have been pursued through efforts both in the soft robots community via morphological computation approaches [4] and in conventional robotics via concurrent design (co-design) strategies [2], [5]. However, existing methods are dominated by addressing the problem of design for a single task, resulting in high-performance task-specialized robots. The main goal of this letter is to provide a co-design framework for robots that are designed for efficiency, while able to use active control for adaptability. Our key insight here is that variability in the environment or task at runtime represents uncertainty at design time. To address this aspect, this letter considers the application of stochastic programming (SP) strategies [6], originally designed for optimization under uncertainty, and demonstrates the benefits of their use for solving co-design problems.

To ensure a versatile design, the proposed approach samples potential scenarios that a robot could face once it is deployed in the world. Each scenario denotes the combination of a task the robot must complete along with the configuration of the environment (e.g., a monopod robot jumping over flat terrain). Within the SP formulation here, design decisions are posed over a probability distribution of scenarios. Using classical SP constructs, the co-design process can be handled in two stages. Morphology parameters (e.g., motor gear ratios, length and mass of links, position of actuators) represent a first-stage decision that must be made independent of the scenarios. Control parameters (e.g., actuator inputs and contact forces) are second-stage decisions that can be tailored to each scenario. Accordingly, new formulations of the co-design problem here are proposed via a two-stage SP problem [6], in which TO is applied to optimize the motion of the robot for each scenario. This approach results in a large-scale nonlinear optimization problem, where the SP formulation is shown to improve solve time and scalability.

A. Related Work

The co-design of morphology and control parameters is expected to increase the performance of robotic systems, and to reduce the effort and resources required to design robots. Automated and interactive strategies have been proposed for reducing the domain knowledge required for design [7], [8]. Similar methods have been applied to co-design of hardware

and software from high-level specifications (e.g., running speed, jump height, gait pattern) [9], [10]. Such automated frameworks support the design process by reducing the number of design iterations and the time required to produce and test a complete system. Although these strategies could likely help designers understand conflicting design pressures when robots need to perform multiple tasks, existing automated methods have not yet considered trade-offs across tasks.

A trend for robot co-design has been applying TO for a single task with motion and morphology parameters as optimization variables. This approach has been successfully applied to identify optimal designs for parallel and series elastic legged robots [11], [12]. In such cases, gear ratios of motors and stiffness parameters of springs were considered as free decision variables, and the co-design strategy selected the best design by minimizing the total cost of transport (i.e., energy consumed normalized by weight times distance traveled). TO has also been applied as part of a task-based optimal design framework for manipulators [13], [14], with the goal of shaping a desired workspace while avoiding singularities. Toward a different goal, Mombaur [15] used co-design and TO to design self-stable human-like running, demonstrating the potential of co-design and optimization to produce robust robots. As a common feature, this prior work in co-design has focused on design for a single scenario.

Other approaches to support the robot design process include methods relying on parametric TO [5] and sensitivity analysis to establish a direct relationship between morphology and control parameters [3]. Both approaches demonstrate that by co-optimizing morphology and control parameters, actuation requirements can be significantly reduced for a single scenario. These promising results motivate the study of using TO and SP for multiple scenarios in this letter.

B. Contribution

The contribution of this work is a robot co-design framework that produces a versatile design by considering multiple tasks and environments during the co-design process. A key feature of the approach is the formulation of a co-design problem through stochastic programming (SP). The SP approach largely decouples the scenarios and provides advantages in terms of computation time and scalability. The results show faster and more energetically efficient trajectories when compared to cases with the morphology parameters fixed or when the multiple tasks are considered sequentially as part of a single multi-phase optimal control problem. To the best of the authors' knowledge, this work represents the first time SP has been studied for co-design.

SP is distinctly different from non-gradient stochastic optimization (SO) strategies such as genetic algorithms. SO applies randomization in the iteration procedure for a deterministic cost. By contrast, SP applies deterministic algorithms to solve an optimization problem where the cost function considers uncertainty. In the co-design approach here, the likelihood of a robot facing each scenario is captured as uncertainty at design time. A probability weight added to a scenario accounts for its rate of occurrence in a multi-scenario setting. This perspective enables the use of a SP formulation and associated solution methods to treat the versatile co-design problem.

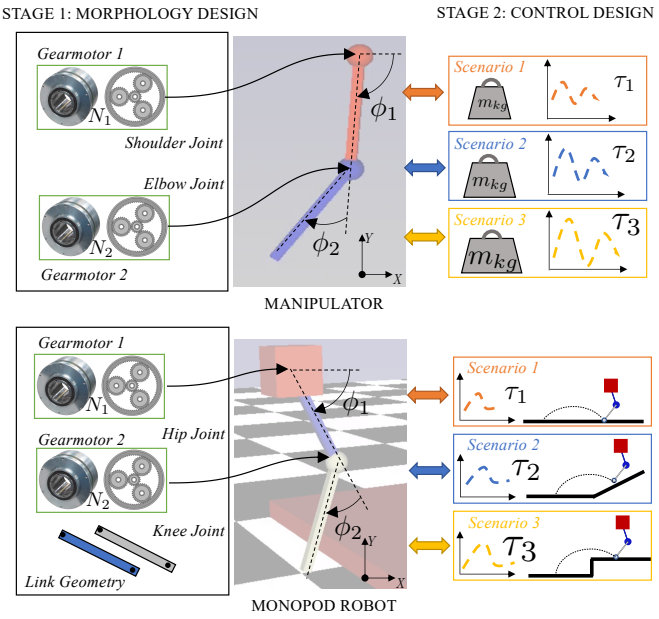

Fig. 1. SP co-design process for a robotic manipulator transporting different loads and a monopod robot jumping across different terrains. Morphology decisions are made at stage one and control decisions at stage two.

The remainder of the letter is organized as follows. Section II provides an overview of SP and describes the new co-design formulation based upon it. Section III presents the main results, shown on a mass-spring-damper system (MSDS), a planar manipulator transporting a range of loads, and a two-link monopod jumping across a range of terrains (Fig. 1). Section IV provides conclusions and discusses future work.

II. Stochastic Programming Co-Design Formulation

To begin, consider the co-design problem for a single scenario. It is assumed that each possible scenario can be parameterized by a vector $\xi \in \mathbb{R}^s$. A single co-design problem is posed as an optimal control problem (OCP)

$$\underset{\mathbf{x}(\cdot),\mathbf{u}(\cdot),\boldsymbol{\rho}}{\text{minimize}} \; V_\xi(\mathbf{x}(\cdot),\mathbf{u}(\cdot),\boldsymbol{\rho}), \tag{1}$$

$$\text{subject to} \; \dot{\mathbf{x}}(t) = \mathbf{f}(t,\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\rho}), \tag{2}$$

$$\mathbf{h}_\xi(t,\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\rho}) \leq \mathbf{0}, \tag{3}$$

$$\mathbf{g}_\xi(t_0,t_f,\mathbf{x}(t_0),\mathbf{x}(t_f),\boldsymbol{\rho}) \leq \mathbf{0}, \tag{4}$$

with decision variables that include state and control trajectories, $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{R}^m$, initial and final times, t_0 and t_f, and design parameters $\boldsymbol{\rho}$. Eq. (1) describes a cost functional $V_\xi(\cdot)$, (2) represents dynamics constraints, (3) path constraints, and (4) describes boundary constraints.

The cost function is expressed in standard form as

$$V_\xi(\mathbf{x}(\cdot),\mathbf{u}(\cdot),\boldsymbol{\rho}) = \ell_{\xi,f}(\mathbf{x}(t_f),\boldsymbol{\rho}) + \int_{t_0}^{t_f} \ell_\xi(\tau,\mathbf{x}(\tau),\mathbf{u}(\tau),\boldsymbol{\rho})\mathrm{d}\tau, \tag{5}$$

where the terminal cost $\ell_{\xi,f}(\cdot)$ and the running cost $\ell_{\xi}(\cdot)$ are user specified and may depend on scenario parameters ξ.

Direct collocation [16]–[18] is applied to transcribe the TO problem (1) into a nonlinear programming problem (NLP). Using direct collocation, a trajectory is divided into spline segments, known as finite elements, and the final solution is approximated by enforcing the constraints at specific collocation points within each element. Similar formulations have been proposed for co-design (e.g., [11], [12]) demonstrating viability of the approach for single-task optimization.

A. Multiple Scenarios via Stochastic Programming

This section poses a co-design problem as an instance of a two-stage SP problem [6]. Co-design for multiple tasks naturally involves different stages of decisions. At the first stage, even before the robot's task/environment is known (e.g., walking vs. jumping), a decision has to be made about the physical parameters of the robot (e.g., the mass and length of its limbs, or the gear ratios for its actuators). At the second stage, a control decision must be made, but can be tailored to the scenario. For instance, periodic inputs might be applied to generate an efficient walking gait, while an aperiodic input might be given for an abrupt jump. The cost of these second-stage decisions is linked to decisions at the first stage, introducing coupling across multiple scenarios.

SP formulations [6] and tailored solutions strategies have been developed to handle multi-stage optimization formulations, where the optimal first-stage decisions are those that minimize the average cost of the best second-stage decisions. Multi-stage SP relies on mathematical programming and probability theory to pose and solve sequential decision-making problems under uncertainty [19]. In contrast to SP, non-gradient stochastic optimization approaches (e.g., evolution algorithms) include randomness in the search procedure.

Consider a set $\Xi \subset \mathbb{R}^s$ that provides a parametric representation of the potential scenarios that a robot could face. For example, if the robot needs to climb various inclines and also to jump obstacles of different heights (as in Fig. 1), Ξ would be a subset of $\mathbb{R}^2$, and each $\xi \in \Xi \subset \mathbb{R}^2$ would encode the slope and the obstacle height of the scenario. Design parameters $\boldsymbol{\rho}$ are optimized for N_s scenarios $\xi_1, ..., \xi_{Ns}$, along with individual state $\mathbf{x}_1(t), ..., \mathbf{x}_{N_s}(t)$ and control $\mathbf{u}_1(t), ..., \mathbf{u}_{N_s}(t)$ trajectories to minimize an overall expected cost

$$\underset{\boldsymbol{\rho},\{\mathbf{x}_i(\cdot),\mathbf{u}_i(\cdot)\}}{\text{minimize}} \quad \sum_{i=1}^{N_s} p(\xi_i) V_{\xi_i}(\mathbf{x}_i(\cdot), \mathbf{u}_i(\cdot), \boldsymbol{\rho})), \tag{6}$$

subject to the constraints (2)-(4) for each scenario, where $p(\xi_i) \in [0, 1]$ is the probability of realization of the scenario encoded by ξ_i. Monte Carlo sampling (from any distribution, e.g., Gaussian, Bernoulli, Uniform) and sparse grids can be used to generate scenarios for SP [20].

In solving (6), although computational requirements per iteration scale polynomially in the number of variables, sparse coupling across scenarios from commonality in the first-stage decision variables can be used to accelerate the solution [21].

B. Dynamic Models: Continuous Dynamics

The dynamic equations in (2) consider planar rigid-body dynamics for each system studied here. Consider the floating-base monopod in Fig. 1. The state $\mathbf{x}(t) = [\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top$ contains the configuration $\boldsymbol{q} \in \mathbb{R}^{n+3}$ with 3 degrees of freedom (DoFs) for the floating base, and n DoFs for the actuated joints. The dynamics take the standard form

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_g = \boldsymbol{S}^\top \boldsymbol{\tau} + \boldsymbol{J}_c^\top \boldsymbol{f}_c, \tag{7}$$

where $\boldsymbol{H} \in \mathbb{R}^{(3+n)\times(3+n)}$ is the joint-space inertia matrix, $\boldsymbol{C} \in \mathbb{R}^{(3+n)}$ gives the Coriolis and centrifugal terms, and $\boldsymbol{\tau}_g \in \mathbb{R}^{(3+n)}$ and $\boldsymbol{\tau} \in \mathbb{R}^n$ denote generalized gravitational forces and actuator torques, respectively. Moreover, $\boldsymbol{S}^\top \in \mathbb{R}^{(3+n)\times n}$ maps actuator torques to generalized forces, and $\boldsymbol{f}_c \in \mathbb{R}^2$ contains external forces with contact Jacobian $\boldsymbol{J}_c \in \mathbb{R}^{2\times(3+n)}$. The dynamics are enforced within the NLP by replicating the steps of the Recursive Newton Euler Algorithm (RNEA) [22].

C. Model Dynamics: Multi-Phase Formulation

To treat hybrid effects for the monopod robot, the TO is further broken into motion phases based on the contact configuration. At transitions between these phases, a modified RNEA-like strategy is used to enforce impulse dynamics due to contact events. To do so, the RNEA forward and backward passes are modified assuming inelastic collisions at a contact point between one foot and the ground.

From the dynamics (7), the net impulse over the interval $\Delta t = t^+ - t^-$ can be addressed by considering the integration

$$\int_{t^-}^{t^+} [\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_g]\, \mathrm{d}t = \int_{t^-}^{t^+} \left[\boldsymbol{S}^\top \boldsymbol{\tau} + \boldsymbol{J}_c^\top \boldsymbol{f}_c\right] \mathrm{d}t, \tag{8}$$

where t^- and t^+ denote the times just before and after impact. From (8), for an infinitesimal time interval, the net change in generalized momentum is given by

$$\boldsymbol{H}(\boldsymbol{q}) \left[\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^-\right] = \boldsymbol{H}(\boldsymbol{q})\Delta\dot{\boldsymbol{q}} = \boldsymbol{J}_c^\top \hat{\boldsymbol{f}}_c, \tag{9}$$

where $\hat{\boldsymbol{f}}_c$ is the impulse force due to impact, and $\dot{\boldsymbol{q}}^-$ and $\dot{\boldsymbol{q}}^+$ are velocities just before and after impact, respectively.

Using $\boldsymbol{H} = \sum_{i=1}^{n} \boldsymbol{J}_i^\top \boldsymbol{I}_i \boldsymbol{J}_i$ substituted into (9) produces

$$\sum_{i=1}^{n} \boldsymbol{J}_i^\top \boldsymbol{I}_i \Delta\boldsymbol{v}_i = \boldsymbol{J}_c^\top \hat{\boldsymbol{f}}_c, \tag{10}$$

where $\boldsymbol{I}_i \in \mathbb{R}^{3\times3}$ is the spatial inertia of the i-th body [22], and $\Delta\boldsymbol{v}_i$ is its change in spatial velocity due to impact. $\Delta\boldsymbol{v}_i$ is enforced in the TO using a modified RNEA-like strategy. The conventional RNEA provides $\boldsymbol{\tau} = \mathrm{RNEA}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{f}_c)$, whereas the contact event must satisfy $\mathbf{0} = \mathrm{RNEA}(\boldsymbol{q}, \mathbf{0}, \Delta\dot{\boldsymbol{q}}, \hat{\boldsymbol{f}}_c)$. The significance of the zero on the left side is that the generalized impulse delivered by the actuators must be zero at impact.

Another aspect of mutli-phase optimization is the need for flexibility in the timing of each motion phase. Switching times t_j are included as optimization free variables, with switching constraints added at the end of each phase. Consider a set of n_{ph} motion phases, with the time interval for phase j denoted $[t_{j-1}, t_j]$. The length of phase j is denoted as $T_j = t_j - t_{j-1}$.

Through a change of variables to address timing optimization, the total running cost for one phase takes the form

$$\int_0^1 T_j \, \ell(t' T_j, \mathbf{x}'_j(t'), \mathbf{u}'_j(t')) \mathrm{d}t', \tag{11}$$

and modified dynamics are expressed by

$$\frac{\mathrm{d}\mathbf{x}'_j(t')}{\mathrm{d}t'} = T_j \mathbf{f}(\mathbf{x}'_j(t'), \mathbf{u}'_j(t')), \tag{12}$$

with $t' \in [0, 1]$ as a scaled version of time, and the $'$ superscript denoting time-scaled variables.

III. Co-Design Results

Co-design was carried out for three systems. A linear mass-spring-damper system (MSDS) served to illustrate concepts of co-design via TO and SP. Next, SP co-design was applied for a two-link robotic manipulator and a planar monopod. The framework was implemented using the Python optimization modeling language Pyomo [23], [24] with PySP [25], and solved using IPOPT [26] on a 3-GHz Intel Core i7 processor.

A. Mass-Spring-Damper System

The first co-design problem considered is for a MSDS controlled by an external force. The system has a 1.0 [kg] mass m, spring constant k, and damping coefficient b. The system is connected to a vertical wall and slides across a frictionless surface such that its dynamics take the form $m\ddot{x}(t) + b\dot{x}(t) + kx(t) = F(t)$, with the force $F(t)$ treated as a control input.

A desired trajectory is specified as: $x_d(t) = \overline{x} + A\sin(\omega t)$, where the position mean value $\overline{x}$, amplitude A, and frequency ω provide a parametric representation of the scenarios the system could face. The co-design problem is stated as follows: *Find the best fixed stiffness k and damping b along with input forces $F(t)$ to minimize the tracking error $e(t) = x_d(t) - x(t)$ across a range of desired trajectories $x_d(t)$.*

The cost functional for the optimization was chosen as a weighted sum of the tracking error and input force:

$$V(\cdot) = \int_0^{t_f} \left(w_e \|e(t)\|^2 + w_F \|F(t)\|^2 \right) \mathrm{d}t, \tag{13}$$

where w_e and w_F are weights for the error and the force set to $10000/\overline{x}^2$ and $10/(mg)^2$, respectively, with g denoting the gravitational acceleration. Path and boundary constraints included: $|x(t)| \leq 10$ [m], $F(t) \geq 0.0$ [N], $k \geq 0.1$ [N/m], and $b \geq 0.1$ [N $\cdot$ s/m].

To create scenarios, the probability of each parameter was assumed to follow a normal distribution: $\overline{x}(\xi) \sim \mathcal{N}(2.0, 0.5)$, $A(\xi) \sim \mathcal{N}(1, 0.5)$, $\omega(\xi) \sim \mathcal{N}(0.0, 3.0)$. All parameters were assumed independent, and scenarios ξ_i were chosen on a grid with 8 equidistant points in each dimension. Probability weights $p(\xi_i)$ arose from normalizing the probability-density values across all scenarios (i.e., dividing individual probability densities by the sum for all scenarios so that $p(\xi_i) \in [0,\ 1]$).

Table I summarizes the results from SP co-design with 8 and 512 scenarios, and single-task co-design for the best- and worst-case scenarios of the 512 set (i.e., lowest and highest values for V_ξ at the optimum of (6)). Co-design for the best- and worst-case scenarios highlights conflicting design pressure across these cases. For instance, the optimal k for the worst-case alone was almost 5 times bigger than for the best-case.

Fig. 2 shows the tracking error for the system following co-design with 512 scenarios. Highlighted in blue and red are respectively the results for the best- and worst-case scenarios. The bottom plot shows the distribution of the cost V_{ξ_i} and weighted cost $p(\xi_i)V_{\xi_i}$ for each scenario listed in ascending order of scenario probability $p(\xi_i)$. SP effectively emphasizes performance of the most-likely scenarios: a high weighted cost indicates that design pressure is allocated to a scenario, while a low (non-weighted) cost indicates high performance. This strategy, naturally, runs the risk of poor performance in the less probable cases [20]. To illustrate, if the design was tailored to the worst-case scenario, by applying single-task co-design

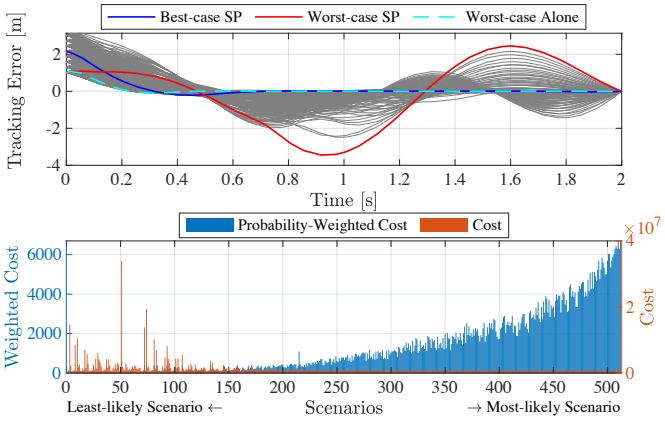

Fig. 2. Mass-spring-damper system: (Top) Open-loop tracking error from optimization with 512 scenarios. Results in gray show scenarios in between the best- and worst-case scenarios. The cyan dashed line marks the tracking error for the worst-case scenario optimized alone. (Bottom) Cost and probability-weighted cost with scenarios listed in ascending order of scenario probability.

TABLE I
Co-Design Results for Tracking Control with the MSDS (10 finite elements, 5 collocation points per element.)

Problem	Morphology		Motion-Control	IPOPT		
	k [N/m]	*b [N·s/m]*	*Max. Force [N]*	*Variables*	*CPU Time [s]*	*Weighted Avg. Cost*
Best-case Scenario	5.67	4.34	263.15	512	0.23	1,256.08
Worst-case Scenario	26.30	1.44	456.38	512	0.80	1,900.61
SP 8 Scenarios	7.12	3.81	292.24	4,082	4.781	1,245.18
SP 512 Scenarios	8.83	2.51	309.71	1,240,064	518.529	1,345.95

optimization, significantly lower tracking error is achieved, as the cyan dashed line in the upper plot of Fig. 2 shows.

From Table I, the results from SP co-design with 8 and 512 scenarios show that as the number of scenarios increases, the design is better at allocating control power to follow a desired trajectory. To illustrate, the natural frequency for the 512- and 8-scenario designs are $2.97\,[\mathrm{rad/s}]$ and $2.67\,[\mathrm{rad/s}]$, respectively. The 512-scenario configuration presents a natural frequency closer to the average frequency ($3\,[\mathrm{rad/s}]$) for all the desired trajectories. Beyond the 512-scenario limit, the system natural frequency displayed convergence to $3\,[\mathrm{rad/s}]$. This result suggests that the co-design formulation is able to tune the passive dynamics of the system to handle average cases, while relying on control to adapt across scenarios.

B. Two-link Robotic Manipulator

The second co-design problem considered a planar two-link manipulator required to transport a range of loads across its workspace. The manipulator consists of a serial 2-link mechanism connected by revolute joints. The mass (1 [kg]) and length (0.5 [m]) of the links are fixed parameters, while the gear ratios at the motors are optimized parameters. The co-design problem is stated as follows: *Find the best combination of motor gear ratios N_i and torque inputs $\tau_i(t)$ for the manipulator to complete a range of load transport tasks with minimum energy consumption.*

The same cost functional below was applied for the manipulator and for the monopod in Section III-C. The objective includes three weighted terms. The first term penalizes the task-completion time. The second term considers the total electrical energy required to power the motors, with p_e denoting instantaneous electrical power. The third term penalizes joint forces $\mathbf{f}_i$, accelerations $\ddot{\boldsymbol{q}}$, and velocities $\dot{\boldsymbol{q}}$, to produce smooth motions, where i denotes the joint number.

$$V(\cdot) = T^2 + T\int_0^1 p_e(t)\,\mathrm{d}t + T\int_0^1 \left(w_v\|\dot{\boldsymbol{q}}(t)\|^2 + w_a\|\ddot{\boldsymbol{q}}(t)\|^2 + w_f\sum_i \|\mathbf{f}_i(t)\|^2 \right)\mathrm{d}t. \tag{14}$$

Weights w_v, w_a and w_f for the joint velocities, accelerations, and forces were set as 0.0, 0.5, and 0.1, respectively. Weight coefficients serve as user-defined tuning parameters that help the search avoid undesirable local optima via regularization. The need for these regularizing terms is highlighted in other co-design work as well [2], [5]. To fully specify the optimization problem, path and boundary constraints included: $-3\pi/2 \leq q_1 \leq \pi$ [rad], $|q_{i\geq 2}| \leq 7\pi/8$ [rad], $|\tau_i| \leq 100$ [$\mathrm{N\cdot m}$], and $N_i \geq 1.0$.

The electrical power was computed assuming Emoteq HT-5001 frameless motors with 100% regeneration efficiency for simplicity. The mass of each motor was taken as 1.28 [kg], with an assumed 50% mass increase over data-sheet values due to gearbox mass. The electrical power for each motor was modeled as

$$p_{e_i} = \tau_{m_i}\dot{\theta}_{m_i} + \frac{\tau_{m_i}^2}{k_m^2}, \tag{15}$$

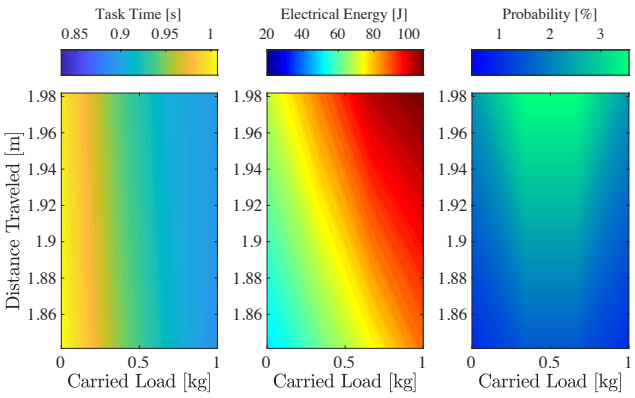

Fig. 3. Two-link manipulator: Colormaps relating distance traveled and carried load to (left) task time, (center) electrical energy, (right) scenario probability. Results for 64 scenarios.

where $\tau_{m_i} = \tau_i/\eta N_i$ and $\dot{\theta}_{m_i} = N_i\dot{\theta}_i$ are the torque at the motor shaft, and the motor angular velocity, respectively, $k_{m_i} = 0.46[Nm/\sqrt{W}]$ denotes the motor torque efficiency constant, and $\eta = 0.75$ is an assumed gearbox efficiency. The operation range of the motors was limited by the maximum winding voltage $V_{max} = 100\,[V]$:

$$\left|\sqrt{R_{m_i}}\left(\frac{\tau_{m_i}}{k_{m_i}} + k_{m_i}\dot{\theta}_{m_i}\right)\right| \leq V_{max}, \tag{16}$$

where $R_{m_i} = 6.82\ [\Omega]$ is the motor terminal resistance.

Parameters for each scenario ξ_i were determined using initial joint positions following a normal distribution $q_1(0) \sim \mathcal{N}(-\pi/2, \pi/4)$, $q_2(0) \sim \mathcal{N}(0.0, \pi/8)$. In addition, the load carried by the end-effector was taken with Gaussian probability density $m_{load} \sim \mathcal{N}(0.5, 0.25)$. Scenarios were chosen on a grid with 4 equidistant points in the $q_1(0)$, $q_2(0)$, and m_{load} dimensions. Final joint positions were set with respect to the initial positions as $q_1(t_f) = q_1(0) + \pi$ and $q_2(t_f) = -q_2(0)$.

Co-design was run for up to $64 = 4^3$ scenarios, with 15 finite elements per scenario, and 5 collocation points per element. Fig. 3 shows the optimized task-completion time and electrical energy required for SP co-design with 64 scenarios. Higher energy expenditure is associated with more challenging tasks (heavier loads and higher distances traveled). From the left subfigure, as the load increases, the task time is reduced by applying higher torques at each joint. By contrast, with lightweight loads, the system relies more on its free dynamics, which increases the task-completion time.

To assess the SP formulation, the SP co-design approach (SP Co-Design) was compared to a sequential approach (Sequential Co-Design), and to a fixed-design approach (SP Fixed Design). As part of this assessment, the robot performed three consecutive tasks involving the following operations: (1) swing-up motion carrying a 1.0 [kg] load, (2) repositioning motion without load, and (3) lateral motion carrying a 1.0 [kg] load. The sequential co-design approach optimized the three tasks through a multi-phase formulation, while the SP co-design approach optimized each task trajectory individually, keeping the gear-ratio design coupled among tasks. For the SP fixed-design approach, the gear ratios were found in advance from co-design optimization for the first task alone.

By contrast to the sequential method, the SP formulation introduces additional sparsity to the problem by not connecting each scenario trajectory to one another. Compared to a sequential co-design strategy, even if the trajectories are not connected in a multi-phase formulation but are solved as part of a monolithic optimization problem, the SP formulation provides equal design flexibility and shows to be more efficient. The SP solution is carried out by PySP [25] in Pyomo [23].

Results for the robotic manipulator are summarized in Table II. The results point to a 78% computation-time reduction when using SP Co-Design rather than the sequential formulation. For the 3-task experiments, 10 finite elements and 7 collocation points were used. Beyond this limit, the sequential formulation became intractable, with high iteration times (up to 71 seconds per iteration when using 20 finite elements and 5 collocation points). This result showcases the benefit of the SP formulation for scalability to find higher accuracy solutions, and to handle more scenarios.

According to Table II, the design obtained using SP Co-Design used smaller gear ratios and required smaller torques than either the fixed design or the design from the sequential formulation. Moreover, design flexibility due to co-design led to smaller CPU times compared to a fixed-design strategy. In terms of performance, SP Co-Design produced the fastest and the best overall design. To support the results in Table II, Fig. 4 shows joint-angle, joint-torque, and electrical-energy profiles for the three tasks with the 2-link manipulator. In general, the SP formulation found faster profiles ($\approx 50\%$ overall time reduction), which led to less energy expenditure. It is noted how the sequential formulation reaches the lower torque limit (-100 [N·m]) to complete the second task, which signals a nonsmooth behavior compared to the SP results.

To further test scalability, the co-design problem was solved for the manipulator performing the same three tasks but with a varying number of links. A fixed total mass (5.84 [kg] including links and motors) and total length (1 [m]) was imposed, matching the totals for the previous 2-link manipulator. Mass and length were evenly distributed for each link, and the joint angles for the second and third tasks were assigned so that the links formed the upper half of a regular polygon. Fig. 5 provides the computation time and the time per iteration from the co-design problem solved using IPOPT and the SP framework. As a highlight, the 3-task co-design problem for a 16-DoF manipulator was solved at 2.02 seconds per iteration,

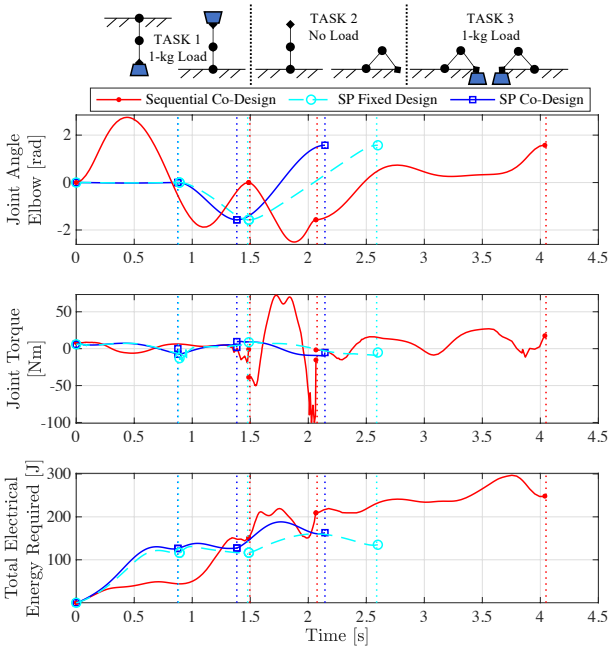

Fig. 4. Two-link manipulator performing 3 consecutive tasks as depicted by the drawings above the legend. The dotted lines mark the task-completion time. The plots show: (top) joint angle and (middle) torque at second/elbow joint, (bottom) total electrical energy required.

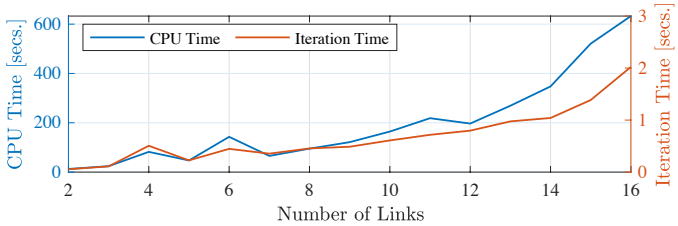

Fig. 5. SP-framework Scalability: IPOPT CPU time and time per iteration for an n-link robotic manipulator performing three load-transportation tasks.

while using the sequential formulation, the co-design problem could not be solved with more than 3 links. The results in Fig. 5 support future use of SP for co-design of systems such as industrial manipulators (≈7 DoFs) or quadruped robots (≈12 actuated DoFs). For comparison, the state of the art reports co-design implementations [3] for a single task with a 3-DoF manipulator and a quadruped (14 actuated DoFs) with iteration times of 1.5 and 32.1 seconds respectively.

TABLE II
CO-DESIGN RESULTS FOR LOAD TRANSPORTATION TASKS WITH A TWO-LINK MANIPULATOR.

Problem	**Morphology**		**Motion-Control**		**IPOPT**				
	Gear Ratios		***Max. Torque [N·m]***		***Variables***	***CPU Time [s]***	***Non-Weighted Cost***		
	N_1	N_2	τ_1	τ_2			Task 1	Task 2	Task 3
Sequential Co-Design 3 Tasks	13.20	6.56	86.94	100.00	12,789	121.65	12,818.05	9,473.56	11,357.74
SP Fixed Design 3 Tasks	15.44	17.36	86.00	9.28	13,211	136.63	733.53	238.14	631.71
SP Co-Design 3 Tasks	12.77	9.45	45.03	9.89	13,217	26.30	742.36	173.35	441.32
								Weighted Avg. Cost	
SP Co-Design 64 Tasks	15.78	10.23	21.23	32.27	301,762	4,738.58		638.98	

C. Planar Monopod Robot

The final co-design problem considers a planar monopod. The robot model has four DoFs $\boldsymbol{q} = \begin{bmatrix} x_{base}, & y_{base}, & \phi_1 & \phi_2 \end{bmatrix}^\top$, and is actuated by two gearbox motors (see Fig.1). The total mass was preliminarily set to 2.0 [kg], without counting the mass added by the motors and gearboxes. The lengths of the links were also preliminarily set to 0.45 [m] for the thigh and 0.55 [m] for the shin. This model exhibits the main difficulties of legged robots (i.e., nonlinear dynamics and contact switching). Such difficulties pose an additional challenge for the SP framework since each scenario is connected to a multi-phase OCP. Nevertheless, the SP formulation still reduces the computational effort.

The co-design problem is stated as follows: *Find the combination of motor gear ratios N_i and torque inputs $\tau_i(t)$ for the robot to jump forward at least* 0.3 *[m] with the least energy consumption over a variety of terrains.* The objective function (14) was set with weights w_v, w_a, w_f of 0.5, 0.0, and 0.1, respectively. Inequality constraints included: $|\phi_1| \leq 7\pi/8$ [rad], $|\phi_2| \leq 3\pi/4$ [rad], $|\tau_i| \leq 200$ [N $\cdot$ m], and $N_i \geq$ 1.0. For each scenario, problem (1) was treated more generally as a multi-phase OCP with contact sequence: (P1) stance, (P1$\rightarrow$P2) impact, and (P2) flight.

Three SP design frameworks were tested: (SP Fixed Design) not optimizing morphology parameters, (SP Partial Co-Design) optimizing gear ratios only, and (SP Full Co-Design) optimizing gear ratios and the mass and length of the links. The optimization was run for three jumping scenarios: (S1) over flat terrain, (S2) onto a stair (height = 0.1 [m]), and (S3) onto an inclined terrain (slope = 25%). Foot contact constraints were enforced using a Coulomb-friction model for constraints on the tangential force F_t and normal force $F_N \geq$ 0.0 [N] according to $|F_t| \leq \mu F_N$. The friction coefficient μ was specified according to a Gaussian distribution $\mu(\xi) \sim \mathcal{N}(0.7, 0.05)$. Designs were required to jump at least 0.3 [m], while there were no constraints on the height. In the full co-design formulation, the mass was constrained based on link length using the density of aluminum and assumed fixed geometric dimensions perpendicular to the lengths.

Results for the SP Partial Co-Design framework are plotted in Fig. 6. Overall, the highest energy consumption occurs for the scenario of jumping onto a stair. Table III summarizes the results for the three design frameworks. To produce results for the SP fixed-design framework, gear ratios were found from co-optimization for the first scenario alone. When using this fixed design for the three scenarios, the optimization solution took almost twice as long as with design flexibility using SP Partial Co-Design. Fig. 7 compares results from the three SP design frameworks for the second scenario only. As more morphological parameters were optimized through co-design, energy use or/and performance (i.e., task completion time) were improved. To illustrate, SP Partial Co-Design improves energy use only, while task-completion time remains similar as with fixed design. In contrast, SP Full Co-Design reduced the task-completion time and the energy use. Including the mass and length of the links in the optimization results in a design that is 30% lighter overall, with $\approx 60\%$ and $\approx 40\%$ reduction in mass and length, respectively. Optimal thigh-shin ratios for the mass (m_{thigh}/m_{shin}) and the length (l_{thigh}/l_{shin}) resulted equal to 1.0, which approximates values (1.17 and 1.31) reported for two-link legged quadruped walking [2].

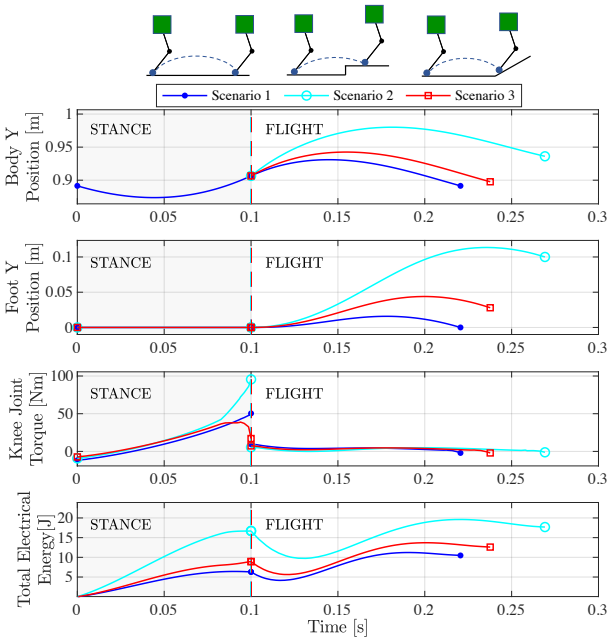

Fig. 6. Performance of monopod robot for three jumping scenarios. The dashed lines mark the transition from stance to flight. The plots show: (Top) body height, (second from top) foot height, (third from top) knee-joint torque, (bottom) total energy energy required.

To further validate the results in terms of energy use, the cost of transport (CoT) was also computed. Average values across scenarios are: SP Fixed Design CoT $= 1.38$, SP Partial Co-Design CoT $= 0.79$, SP Full Co-Design CoT $= 0.70$. Co-design reduces the CoT, and augmented design flexibility leads to higher CoT reduction.

TABLE III
CO-DESIGN RESULTS FOR JUMPING TASKS WITH A MONOPOD ROBOT. (15 FINITE ELEMENTS PER PHASE, 4 COLLOCATION POINTS PER ELEMENT.)

Problem	**Morphology**				**Motion-Control**		**IPOPT**				
	Thigh-Shin Ratios		***Gear Ratios***		***Max. Torque [N·m]***		***Variables***	***CPU Time [s]***	***Non-Weighted Cost***		
	Length	Mass	N_1	N_2	τ_1	τ_2			Task 1	Task 2	Task 3
SP Fixed Design	0.82	0.67	9.61	17.01	70.52	70.05	27,887	271.47	111.51	203.85	125.31
SP Partial Co-Design	0.82	0.67	12.85	15.89	79.85	95.55	27,899	156.97	107.94	198.13	134.88
SP Full Co-Design	1.00	1.00	10.08	13.73	62.97	75.94	27,915	215.45	69.83	151.50	95.62

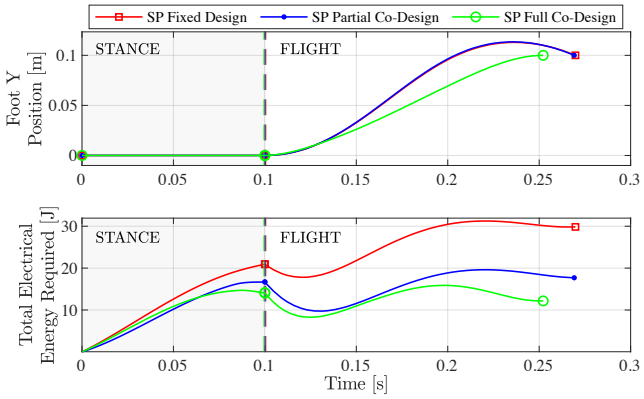

Fig. 7. Co-design of a monopod robot jumping onto a stair. Results in green portray the case when the mass and length of the links are optimized in combination to the motor gear ratios and motor inputs: (top) foot height, (bottom) electrical energy. Dashed lines mark phase transitions.

IV. Conclusions and Future Work

The co-design framework presented builds the requirement of design versatility into the problem formulation. This approach is enabled by the use of SP, which treats uncertainty in the task/environment at design time through a two-stage formulation [6]. Stochasticity and probabilistic reasoning come in place when the scenarios considered are assigned weights from a probability measure over the events [19]. The use of PySP [25] enables the structure of this problem to provide accelerated solution strategies. Three case studies have shown that the two-stage SP characterization for optimizing design and control more efficiently addresses multi-scenario requirements compared to sequential formulations, while offering an automated way to balance performance across multiple scenarios.

This work has shed led on many areas of potential improvement for SP co-design. With the proposed formulation, extreme scenarios with low probabilities add risk to the design. The use of risk measures as part of the formulation, such as the conditional value at risk (CVaR), represents a future prospect for the method. Furthermore, dealing with hybrid formulations for legged robots still imposes a computational challenge for the co-design framework. To accelerate convergence, future work will explore the use of progressive hedging algorithms (PHA) [27] and decomposition algorithms [28], [29] that are tailored to accelerating the solution of large-scale SP problems.

Acknowledgment

The authors gratefully acknowledge Alex Dowling for introducing them to SP and the available supporting tools.

References

[1] T. McGeer, "Passive dynamic walking," *International Journal of Robotics Research.*, vol. 9, no. 2, pp. 62–82, 1990.

[2] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Task-based limb optimization for legged robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 2062–2068.

[3] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1521–1536, 2018.

[4] R. Pfeifer and G. Gómez, *Morphological Computation - Connecting Brain, Body, and Environment*. Springer Berlin Heidelberg, 2009, pp. "66–83".

[5] A. Spielberg, B. Araki, C. Sung, R. Tedrake, and D. Rus, "Functional co-optimization of articulated robots," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 5035–5042.

[6] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[7] R. Desai, Y. Yuan, and S. Coros, "Computational abstractions for interactive design of robotic devices," in *IEEE International Conference on Robotics and Automation*, May 2017, pp. 1196–1203.

[8] R. Desai, B. Li, Y. Yuan, and S. Coros, "Interactive co-design of form and function for legged robots using the adjoint method," in *International Conference on Climbing and Walking Robots*, 2018, pp. 125–133.

[9] A. M. Mehta, J. DelPreto, B. Shaya, and D. Rus, "Cogeneration of mechanical, electrical, and software designs for printable robots from structural specifications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2892–2897.

[10] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, no. 6799, pp. 974–978, 2000.

[11] Y. Yesilevskiy, W. Xi, and C. D. Remy, "A comparison of series and parallel elasticity in a monoped hopper," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 1036–1041.

[12] Y. Yesilevskiy, Z. Gan, and C. D. Remy, "Optimal configuration of series and parallel elasticity in a 2d monoped," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1360–1365.

[13] C. J. J. Paredis and P. K. Khosla, "Kinematic design of serial link manipulators from task specifications," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 274–287, 1993.

[14] R. E. Stamper, Lung-Wen Tsai, and G. C. Walsh, "Optimization of a three DOF translational platform for well-conditioned workspace," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, April 1997, pp. 3250–3255 vol.4.

[15] K. Mombaur, "Using optimization to create self-stable human-like running," *Robotica*, vol. 27, no. 3, p. 321330, 2009.

[16] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[17] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010, vol. 10.

[18] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010, vol. 19.

[19] B. Defourny, D. Ernst, and L. Wehenkel, "Multistage stochastic programming: A scenario tree based approach to planning under uncertainty," in *Decision theory models for applications in artificial intelligence: concepts and solutions*. IGI Global, 2012, pp. 97–143.

[20] J. A. Renteria, Y. Cao, A. W. Dowling, and V. M. Zavala, "Optimal pid controller tuning using stochastic programming techniques," *AIChE Journal*, vol. 64, no. 8, pp. 2997–3010, 2018.

[21] C. G. Petra, O. Schenk, M. Lubin, and K. Gärtner, "An augmented incomplete factorization approach for computing the schur complement in stochastic optimization," *SIAM Journal on Scientific Computing*, vol. 36, no. 2, pp. C139–C162, 2014.

[22] R. Featherstone and D. Orin, "Chapter 2: Dynamics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer, 2008.

[23] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola, *Pyomo - optimization modeling in python*. Springer, 2017, vol. 67.

[24] B. Nicholson, J. D. Siirola, J.-P. Watson, V. M. Zavala, and L. T. Biegler, "Pyomo.dae: A modeling and automatic discretization framework for optimization with differential and algebraic equations," *Math Programming Computation*, vol. 10, no. 2, pp. 187–223, 2018.

[25] J.-P. Watson, D. L. Woodruff, and W. E. Hart, "PySP: modeling and solving stochastic programs in Python," *Math Programming Computation*, vol. 4, no. 2, pp. 109–149, 2012.

[26] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[27] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of operations research*, vol. 16, no. 1, pp. 119–147, 1991.

[28] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[29] J. R. Birge, "Decomposition and partitioning methods for multistage stochastic linear programs," *Operations research*, vol. 33, no. 5, pp. 989–1007, 1985.