2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Computational Design of Closed-Chain Linkages: Hopping Robot Driven by Morphological Computation

Kirill V. Nasonov[1], Dmitriy V. Ivolga[1], Ivan I. Borisov[1], and Sergey A. Kolyubin[1]

***Abstract*— The main advantages of legged robots over wheeled ones are their abilities to traverse on uneven terrain due to the use of intermittent contacts and an ability to shift the center of mass relative to the contact location. A robot's leg design can be implemented by using an open-chain mechanism actuated with high-density torque actuators though this solution needs a vast energy budget. An alternative way to design a leg mechanism is the application of morphological computation principle. According to the principle, most of the desired robot's behavior can be delegated to the mechanics with minimum control effort needed to excite, stabilize or augment it. Within this paper, we have proposed a method to synthesize a leg for hopping robots. Due to optimization of mechanical structure, geometric parameters, mass distribution, and elasticity allocation, our method allows getting an energy-efficient robot with minimal control system complexity, which is accomplished via series elastic allocation and active variable length link. Based on this approach, we have designed a hopping robot with two low performance actuators that can achieve hopping, running, and, in the case of a biped or quadruped robot, walking motion. The paper describes a synthesized leg linkage and overviews prototype design, control strategy, and test results of a physical prototype.**

I. Introduction

In the last decade, there has been rapid progress in different fields of robotics focused on physical interaction: low-level control, trajectory planning, hardware, actuators. However, current generation robots are still far from the robustness, efficiency, and intelligence as those are expected by the general public. Today's robots are still brittle in the case of an unstructured, a priori unknown, dynamical environment [1]. Future generation of robots is expected to be able to 'exit' well-determined industrial spaces and operate in an undetermined environment initially built for human beings.

We have already witnessed rapid progress in computer vision, natural language processing, control, trajectory generation, driven by machine learning (ML). Thus, there are high expectations on machine learning to accelerate robots' evolution. However, applying learning methods to embodied, physical robots is challenging since they are intended to act and interact in the real world and within physical constraints imposed thereby. Nevertheless, ML techniques such as reinforcement learning are successfully used, e.g., to generate joints' trajectories.

*This work was supported by the Analytical Center for the Government of the Russian Federation (IGK 000000D730321P5Q0002), agreement No. 70-2021-00141.

[1]Kirill V. Nasonov, Ivan I. Borisov, Dmitriy V. Ivolga, and Sergey A. Kolyubin are with the Biomechatronics and Energy-Efficient Robotics Lab, ITMO University, Saint Petersburg, Russia e-mail: {kvnasonov, borisovii, s.kolyubin}@itmo.ru

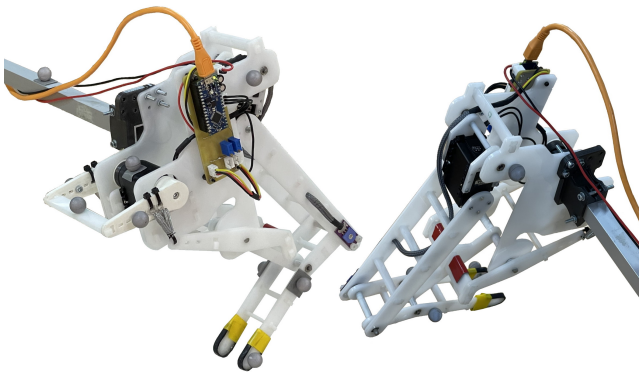

Fig. 1. Images of a physical prototype of an underactuated hopping robot designed to validate the feasibility of the proposed method, physical reproduction and rationale of the synthesis results

From a learning perspective, robots are embodied agents, which exchange energy and information with physical environment. A robot's morphology affects its performance [2]. Controllers' computation power, sensors' accuracy, actuators' capabilities, mechanisms' degrees of freedom and constrains, mechanics compliance, and the whole robot's dynamics of motion have tremendous influence on what the agent can and cannot do.

A. Related Work

Among all possible robots' morphologies, the fully-actuated open chain mechanics is the most popular. Abilities to move in an arbitrary direction, to accurately follow desired trajectories, to physically interact with the environment, to perform planar and spatial motion, make fully-actuated serial kinematics a versatile tool to perform most of desired motion. Despite the apparent advantages, the need for a large number of actuators to steer this kind of design entails problems in terms of actuators' and sensors' performance, control complexity, energy consumption, weight and dimensions, and overall cost.

Modern tools for modeling, control strategies, hardware and materials allow to construct high performance robots, especially, for fully actuated systems with rigid links. However, since the available technologies shape robots' performance, the current generation of quadruped legged robots such as [3], [4], [5] have evolved into robots with similar type of morphology such as a brick-shaped body and four 3R legs, high torque density actuators [6], and control strategy, such as model predictive controllers [7].

Recent studies in the field of soft robotics have shown a paradigm shift for robot design, in which "precision through rigidity" is replaced by "cognition through compliance" [8], [9]. Soft and elastic robots are capable to interact with an unstructured environment, delegating the burden of control from brain to body through embodied cognition. According to an alternative approach, which is called *morphological computation* of control laws (physical intellect embodiment) [10], [11], [12], the major portion of robots' desired behavior can be achieved with the "body" instead of the "brain". It is possible to "program" the robot to perform the task at the mechanical level by synthesizing a closed-chain mechanism with optimization of (*i*) structure and geometric parameters, (*ii*) optimization of mass distribution between links, and (*iii*) optimization of elasticity distribution.

Meanwhile, algorithmic computational control laws are needed to inject energy into the system to augment, excite, and stabilize natural dynamics. Moreover, control laws complexity is set to a necessary minimum to keep natural dynamics controllable [13]. This principle also gives us an 'infinite control bandwidth,' which is a huge advantage if we talk about physical interaction in an unstructured environment, where software-level sensory-motor coordination might not be fast enough.

Morphological computation allows designing efficient robots, but the design process itself is non-formalized. Although there are guidelines to design, the overall process by its very nature is rather art than science. A way to formalize and structure the design process is bionics, mimicking the principles of wildlife, but this approach limits the range of possible solutions. The process of designing such structures is often informal due to combining components of different physical nature in a robotic system, many topologies and parameters tending to infinity, and heterogeneous criteria for optimal solutions, nonlinear objective functions, etc. The result of a design process strongly depends on the developer's creativity, imagination, engineering intuition, judgment, and experience.

In the absence of accurate analytical models and formalized guidelines, design of robots with morphological computation remains a primary manual process. Recently, efforts to automate their design have centered on co-design of both mechanics and control [14], [15], [16], [17], [18], [19], [20]. However, the studies faced a lack of physical replicability, or rationality of physical reproduction.

B. Overview & Contribution

We present a generalized method for interactive design of robotics with morphological computation as a core for generative design of dynamical systems. The approach allows creating different types of robots for a wide range of tasks. To illustrate the potential of the proposed method we have applied it to develop a series of robotic prototypes of different kinds, e.g., leg mechanism for a hopping robot [21], digits' mechanisms for anthropomorphic grippers [22] and robotic hands [23], and even for a spine module for an exosuit to ensure ergonomics [24].

With this paper we propose generalized guidelines for co-design of linkage mechanisms for robotic purposes. The morphological computation is provided through synthesis of closed-chain linkages and introduction of underactuation via elastic elements. Holonomic constraints allow to restrain mechanism motion within the desired workspace, elastic elements provide adaptability for physical interaction with the environment. To validate the feasibility of the method, physical reproduction of synthesis results, and rationality of the final design, we fabricated (Fig. 1) and tested a physical prototype, and compared its performance with simulation data.

II. Proposed Method Outline

The proposed method of co-design for closed-chain linkages consists of three stages that run sequentially: (1) co-design of fully-actuated open-chain mechanism, (2) structural-parametric synthesis of fully-actuated closed-chain mechanism, and (3) trajectory generation for both fully-actuated or underactuated closed-chain mechanism. All stages can be used both jointly, if we need to synthesize an underactuated mechanism with closed kinematics, and independently, if we want an intermediate result, e.g. only open kinematics.

We focus on the challenge of synthesis of planar linkages. Although mechanisms can involve other components such as gear-trains and cams, planar linkages are arguably the most challenging to design and edit [25]. Performance of a robot's mechanism is concurrently determined by its morphology, interactions, and control signals; thus, by co-design we mean simultaneous optimization of the robot's mechanics and joints' trajectories.

We cast computational design of linkages as an inverse problem: given the desired behavior, we search for a robot's morphology and appropriate trajectories that approximate the target as close as possible. To optimize the parameters of interest, we used MATLAB's Simscape Multibody and genetic algorithm from MATLAB's Global Optimization Toolbox. By morphology we mean the mechanism's structure, links' lengths, mass distribution, and parameters of elastic elements. For the sake of optimization simplification, we do not look for an optimal control strategy and do not tune control parameters, but use idealized built-in controllers able to follow any trajectories in a robot's working space.

The proposed method is general enough and can be used to generate linkages for different kinds of purposes. However, for the remainder of the paper, we restrict our discussion to the synthesis of mechanisms for legged robots, or more specifically to the synthesis of a hopping robot capable of jumping in a place and forward. Fig. 2 shows all the stages of the proposed method for synthesis of a hopping robot.

A. Co-design of open kinematics

At the first step we search for an optimal structure of the open-chain mechanism, links' geometric parameters, and joints' trajectories to provide the required motion. Each of

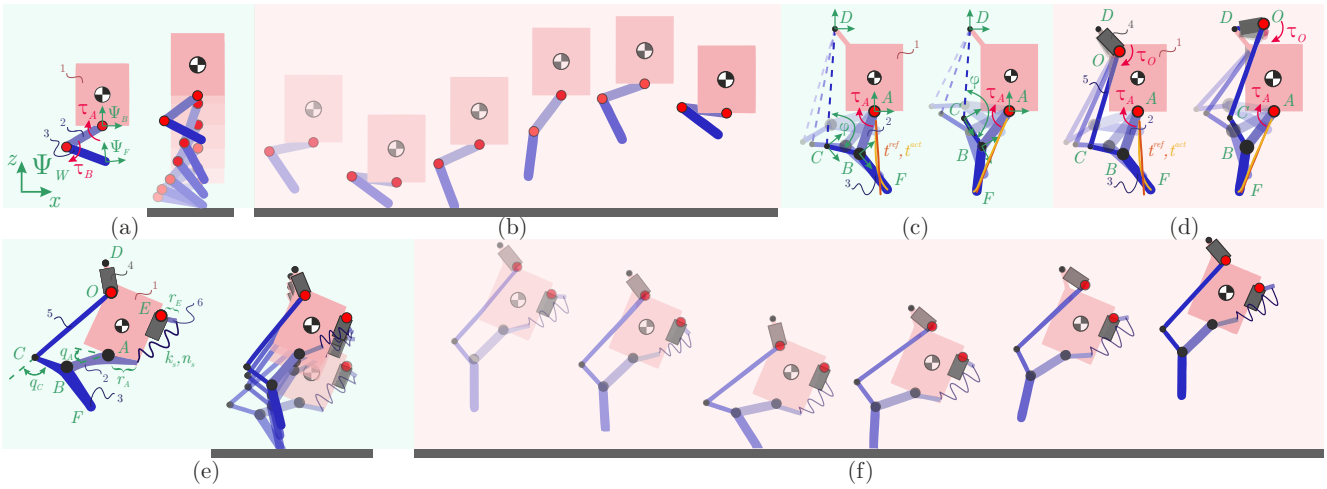

Fig. 2. Design procedure's stages for a leg mechanism: (a) and (b) open-chain mechanism performs jumping in a place and jumping forward behaviors respectively, (c) linkage closure, (d) synthesized closed-chain mechanisms in both modes, (e) jumping in a place sequence and (f) jumping forward sequence for synthesized mechanism with series elastic actuation

the indicated parameters can be predefined a priori or found as a result of an optimization procedure.

Further in the section, the comments given on each aspect are of a fairly general nature. However, for the sake of simplicity of an illustrative example, we have narrowed the space of possible solutions: a morphology open kinematics is given; the task is to find joints trajectories to traverse a flat surface.

1) Input & output data: considering a general case for co-design, a recursive graph grammar can be used to set the optimization procedure [19]. As input data we send a tuple $\mathscr{G}^o_{in}$

$$\mathscr{G}^o_{in} = (N, T, A, R, S),$$

where N and T are sets of non-terminal and terminal symbols respectively, R are production rules, S is a starting symbol, and A is a set of attributes for certain terminal symbols,

$$A = (\mathbf{l}, \mathbf{p}, \mathbf{i}),$$

where $\mathbf{l} \in \mathbb{R}^l$ is a vector of links' lengths, $l \in \mathbb{N}$ is a number of links, $\mathbf{p}$ is a vector of legs' attaching points to a body, and $\mathbf{i}$ is a vector of links' inertia.

As a result of the first stage of synthesis, a fully actuated open-chain mechanism that performs the required movement should be obtained. For each sub-optimal result a tuple $\mathscr{G}^o_{out}$ must be generated

$$\mathscr{G}^o_{out} = (T, A, E, J),$$

where $E \in \mathbb{N}^{b \times b}$ is a structure of connectivity of terminal elements T, $b \in \mathbb{N}$ is a number of mechanism's components, $J(t) \in \mathbb{R}^{l \times m}$ is a structure with joints' trajectories, $m \in \mathbb{N}$ is a number of gaits.

For clarity of the illustrative example, the co-design problem of open kinematics is simplified to the trajectory generation for a linkage with two pin joints. Moreover, to avoid the balance problem, robot's body is allowed to move along $\hat{x}$ and $\hat{z}$ axes without pitch rotation. However, the problem of trajectory generation is still computationally expensive. To efficiently solve the problem, we have considered only the straight line trajectories of a leg's contact point, expressed in the robot's body frame. The leg's contact point trajectory is set as the cubic polynomial trajectory set through multiple waypoints, defined in polar coordinates. Thus, the optimization problem is determined as a task of finding

$$J(t) = f(\mathbf{p}^B_{F\,min}, \mathbf{p}^B_{F\,max}, \alpha^B_F, T),$$

where $\mathbf{p}^B_{F\,min}$ and $\mathbf{p}^B_{F\,max}$ are minimum and maximum lengths between contact point (foot) and the body frame, expressed in the body frame, α^B_F is the corresponding pitch angle, and T is the gait period.

2) Constraints: since optimization is simulation based, all undesirable scenarios can be avoided by setting conditions for simulation abortion. For the illustrative example we set boundary conditions for the robot's body position along vertical axis $z_B \in [z_{B\,min}; z_{B\,max})$, and for mean jumping height $\tilde{h}_F \in [\tilde{h_F}_{min}; \infty)$.

3) Objectives: reward functions are used to evaluate design on a given terrain for a specific behavior, and is computed at every time step of simulation. For jump in a place, a reward function is the following

$$f_{\text{Jump}} = \frac{\tilde{h}^2_F \cdot t^2}{1 + |\tilde{x}_B|}, \tag{1}$$

where numerator tends to increase, while denominator tends to decrease; or a reward function for running behavior

$$f_{\text{Run}} = sgn(x_B) \cdot x^2_B \cdot \tilde{h_F}^2 \cdot t^2, \tag{2}$$

where $\tilde{h}_F$ is the mean jumping height for a foot (contact point), t is duration of simulation, x_B and $\tilde{x}_B$ are an absolute and a mean value of covered distance along $\hat{x}$ axis for a body, respectively. The square root is needed to ease search for the algorithm. Undesirable behavior stops simulation, thus to get the maximum possible simulation time t the robot must not satisfy stop conditions.

4) Optimization: The objectives and constraints described above are functions of the time-varying configuration of a robot's design candidate. We formulate a maximization problem where we seek data for $\mathscr{G}^o_{out}$ that satisfy

$$f(\mathscr{G}^o_{out}) = \max \{ k_j \cdot f_{\text{Jump}} + k_r \cdot f_{\text{Run}} \}, \quad (3)$$

where the coefficients k_j and k_r are weights for the editing objectives penalties.

For the illustrative example we used genetic algorithms to optimize the considered parameters. Fig. 2, (a) and (b) show the animation sequence for the open-chain mechanism.

B. Synthesis of closed kinematics

At the second stage we look for a topology of link groups to be attached for mechanism closure, search for connection points to the initial open-chain kinematics and optimize links' length. The aim is to obtain a fully actuated closed-chain mechanism with the required number of motors.

The optimization procedure of the second stage is similar to the first one; here we specify the main differences.

1) Input & output data: for a general case, a recursive graph grammar can be used to set the optimization procedure. As input data we send a tuple $\mathscr{G}^c_{in}$

$$\mathscr{G}^c_{in} = (\mathscr{G}^o_{out}, N, T, A, R),$$

where $\mathscr{G}^o_{out}$ describes the initial open kinematics, N and T are sets of non-terminal and terminal symbols respectively for links to be attached, R are production rules, and A is a set of attributes for certain terminal symbols, which is analogues to the previous case.

The result is a closed-chain linkage that performs the desired motion using the required number of motors. Similarly to the first stage, each sub-optimal result is described in a form of a tuple $\mathscr{G}^c_{out}$.

2) Intuition of synthesis procedure: the synthesis essence is similar to [25]: we consider links i and j connected to each other through the intermediary link k or a group of intermediary links, and search for a pair of points p_i and p_j such that the distance between them varies least. If the distance is constant throughout the entire cycle of motion, then points p_i and p_j can be connected by a rigid link. An addition of constraint allows removing an actuator.

3) Objectives: the squared mean distance between the points p_i and p_j per cycle of motion:

$$\tilde{l}^2 = \frac{1}{N} \sum_m^N ||p_i^W(t_m) - p_j^W(t_m)||^2,$$

where $p_i^W(t) \in \mathbb{R}^3$ and $p_j^W(t) \in \mathbb{R}^3$ are position vectors of points p_i and p_j respectively, expressed in world frame Ψ_W, N is the number of discrete measurements.

The objective function is the quadratic loss between the actual distance at time of measurement m and mean value l

$$f_e = \frac{1}{N} \sum_m^N (||p_i^W(t_m) - p_j^W(t_m)||^2 - \tilde{l}^2)^2.$$

4) Optimization: if the mechanism has to perform not a single cyclic trajectory t_1, but a series of cyclic trajectories $t \in \{t_1, t_2, \ldots, t_n\}$, we look for connection points such that for each cyclic trajectory a specific length exists for a rigid link that could be attached to the points found. We formulate a minimization problem where we seek data for $\mathscr{G}^c_{out}$

$$f(\mathscr{G}^c_{out}) = \min \{ k_j \cdot f_e^{\text{Jump}} + k_r \cdot f_e^{\text{Run}} \},$$

where coefficients k_j and k_r are weights.

Fig. 2, (c) shows the closed-chain linkages obtained by attaching the link CD to the body (1) and link (3) at points D and C respectively; only joint A is actuated because of adding a holonomic constrain; the right and left figures show configurations for jumping in a place and running, respectively.

If all trajectories $t \in \{t_1, t_2, ..., t_n\}$ are required, then instead of a rigid link, a group of links can be attached to reconfigure the length between the found points. Fig. 2 (d) shows the linkages with links (4) and (5) instead of a rigid link.

5) Discussion: The resulted linkage has the same degree of actuation and can produce the same trajectories of a contact point as open-kinematics. In other words, the design is capable of performing the same behavior. However, the relocated actuator acts differently, since it does not have to perform an oscillating motion, but to hold a constant position. We can utilize less productive actuator, or even use a mechanical clutch to unload an actuator to save energy. Moreover, motor relocation allows to get rid of its inertia for a leg mechanism.

C. Introduction of underactuation

1) General comments: the synthesis of a closed-chain linkage allows *to reduce requirements for the robot's equipment*, e.g., to use less performance motors, less accurate sensors, and to reduce the number of motors and sensors, i.e. *to reduce energy consumption* of a robot.

However, the location of actuators plays a significant role in closed kinematics. We can go even further and set another optimization task to find actuators allocations and optimize their trajectories for better performance. Here, by actuators we mean not only active motors, but also passive elastic elements that could act as a source of torque. Provided by passive elements underactuation *gives mechanical adaptability*, which is a must for efficient physical interaction, allows *to passively generate torque*, and *to store potential energy* needed for recuperation.

The optimization procedure is similar to the previous stages. As input data we send a tuple $\mathscr{G}^a_{in}$, each sub-optimal result is described in a tuple $\mathscr{G}^a_{out}$. The optimization is simulation-based; constrains are similar to the first stage; objectives for jumping and running behaviors are given in eq. (1) and eq. (2); cost function is given in eq. (3).

2) Illustrative example: for the hopper robot we have decided to find a design, that is able to perform dynamical locomotion using low performance servomotors. To put it another way, to find a design that needs the least input torques for motion.

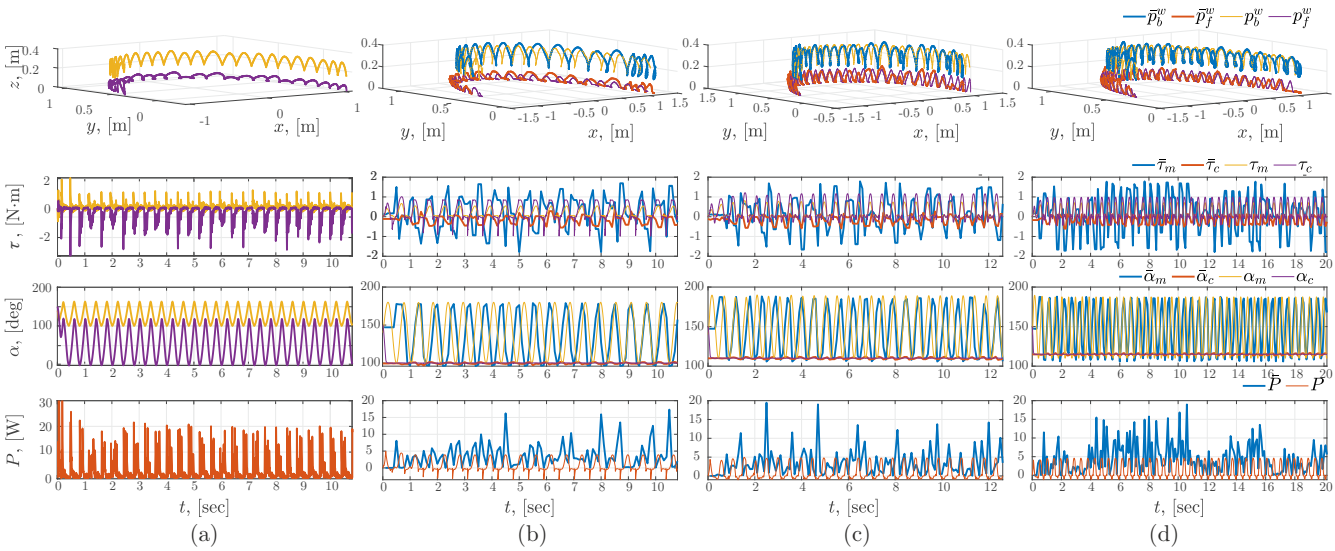

Fig. 3. Both virtual and real experiment data for different robot's configuration: (a) stands for reference open-chain hopping robot, (b) stands for the quickest closed-chain hopping robot motion and (d) for the slowest. p_b^w and p_f^w are positions of body Ψ_B and foot Ψ_F frames respectively expressed in world frame Ψ_W, torques τ and angular potions α for both motors, and overall mechanical power consumption P. Bar sign indicates the experiment data; subscripts m and c indicate the main and the configuration motors respectively. The open-chain robot could not run with the Dynamixels characteristics; to make open-chain robot to run, Dynamixels' torque and velocity were increased by 2 times. Thus, it consumes 4 times more power at touchdowns.

The problem is to find data for tuple $\mathscr{G}^a_{out} = (T, A, E, J)$, where $E \in \mathbb{N}^{b\times b}$ is a structure of connectivity of terminal elements T, $b \in \mathbb{N}$ is a the number of mechanism's components, $J(t) \in \mathbb{R}^{l\times m}$ is a structure with joints' trajectories, $m \in \mathbb{N}$ is a number of gaits, A is a set of attributes for certain terminal symbols, $A = (\mathbf{l}, \mathbf{p}, \mathbf{i}, \mathbf{b})$, where $\mathbf{l} \in \mathbb{R}^l$ contains links' lengths, $l \in \mathbb{N}$ is a number of links, $\mathbf{p}$ is a vector of legs' attaching points to a body, and $\mathbf{i}$ is a vector of links' inertia, $\mathbf{b}$ is a vector of spring parameters.

For the sake of simplification of the illustrative example, we specified the mechanism's topology a priori. To reduce the motor's torque, we have utilized series-elastic actuation concept and set an optimization task to find optimal geometric parameters, such as $\mathbf{l} = (r_E, r_A, q_A)$, and $\mathbf{p} = \mathbf{p}_E^W$, springs parameters $\mathbf{b} = (n_s, k_s)$, and actuators' trajectories $J(t)$ for both motors. The design variables are shown in Fig. 2, (e) along with animation sequences for jumping in a place and running (Fig. 2, f).

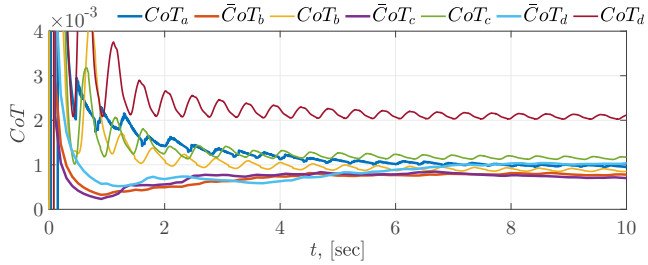

Fig. 4. Costs of transport: (a) open-chain hopping robot, (b) the quickest closed-chain hopping robot motion and (d) the slowest closed-chain. Bar sign indicates the experiment data

III. Prototype Design

To validate the synthesis results of the illustrative example, we have created a physical prototype. The CAD model of the hopping robot has been created according to the found topology and parameters. For ease of manufacturing and assembly, the robot's design was adapted for laser cutting out of polyoxymethylene, which suits best for rapid prototyping. To achieve a lightweight structure, clip connections are used. The axes are lathed from steel 1045. Springs with the most similar parameters to the found ones are chosen.

To actuate the robot, the Dynamixel AX-18A has been chosen, since it is powerful enough and provides a wide range of feedback data. A potentiometer *CJMCU-103* is used to measure knee angle q_C. To control the servos and read sensors, a custom expansion shield was designed on the basis of *Arduino Nano Every*. An interface converter *74LVC2G241* is used to exchange data between the servos and shield. An amplifier *AD623* is used to amplify and filter sensors data. The manufactured prototype weighs 430 g.

A. Virtual Experiment

Before manufacturing, we simulated both open-chain and closed-chain hoppers using MATLAB. Simulation results together with experiment data are shown in Fig. 3 and Fig. 4. CAD model with the detailed physical properties was to decrease *reality gap*. Moreover, we used Simscape Electrical to model Dynamixels and control algorithms. However, since datasheet do not provide all necessary data some parameters, such that inductance, resistance of the windings, inertia and the electromechanical constant were approximately estimated by the parameters of DC motor of similar performance.

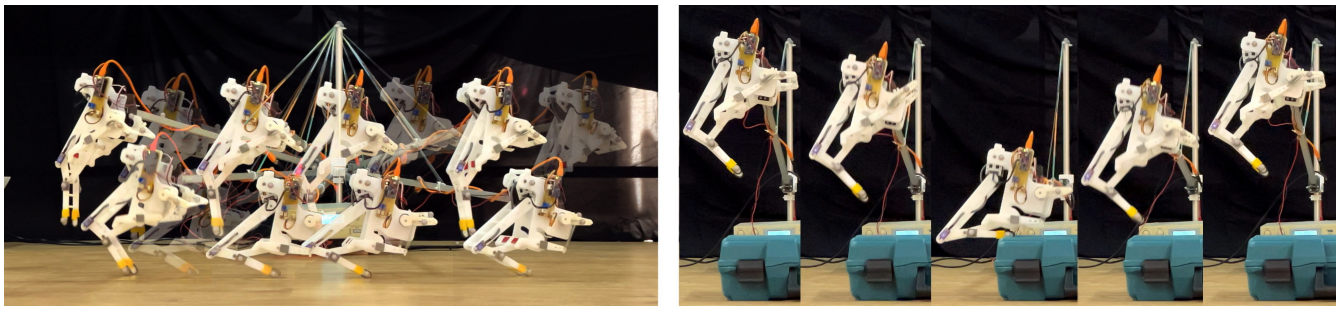

Fig. 5. Sequence of images for running and jumping in a place behaviors. Elastic bands are used to compensate connection rod's inertia

To steer the robot, we use PID controllers to follow the joints' trajectories found during the optimization procedure. To accomplish the resonance based motion, the main motor (joint E) performs oscillating motion to act on springs, while the auxiliary motor (joint O) reconfigures the robot's mode. The leg mechanism folds when the robot hits the ground because of gravity and reaction forces. Due to closed kinematics the link 2 rotates clockwise and stretches the spring; the main motor has to move in the opposite direction to store the maximum potential energy in springs. The stored potential energy is injected back to accomplish the next jump.

Reference trajectory for the main motor $\theta_m^*(t)$ follows

$$\theta_m^*(t) = A^* \cdot \sin(2\pi \cdot \omega \cdot t + q_C) + B,$$

where A^* is the reference amplitude, B is an angular bias for the equilibrium position, q_C is the knee angle, and ω is the actual frequency of oscillation. Fuzzy logic controller adjusts actual frequency ω to follow reference ω^* found during the optimization phase by minimizing error δ

$$\delta = ||\omega^* - \omega||^2 = ||\omega^* - \frac{1}{T}||^2,$$

where T is the actual gait period between touchdowns.

B. Real Experiment

The hopping robot has 2 degrees of freedom (DoF) along the vertical $\hat{z}$ and horizontal $\hat{x}$ axes; to constrain all others DoFs, we rigidly attached a 1.15 meters long connection rod to the robot's body. The another rod's end is connected to a fixed frame via a two pin joints that allows yaw and roll rotations around the fixed base. Fig. 5 shows the sequence of images for running and jumping in a place behaviors. Elastic bands are used to compensate connection rod's inertia.

The Dynamixels AX-18A provides a wide range of feedback data, such as position, angular velocity, and torques. As for the kinematic parameters we have captured the robot motion using eight cameras of real time tracking systems *OptiTrack*. Fig 1 and 5 shows the prototype with reflective spheres. The system's residual mean error equal to 0.9 mm on ray length 5.9 m.

Fig. 3 show data for both virtual and real experiment data (bar sign indicates the experiment data). Tests have been carried out for the robot's different configurations and surface slope. Configuration angle α_c adjusts the trajectory of contact point: motion along vertical line gives jumping in a place behavior, inclination of trajectory leads to the jumping forward behavior. The inclination angle is proportional to the robot's horizontal velocity $\dot{p}_{x_B}^W$. The plots show trajectories for body frame Ψ_B and foot frame Ψ_F expressed in world frame Ψ_W, torques τ and angular potions α for both motors, and overall mechanical power consumption P

$$P = \tau_m \cdot \dot{\alpha}_m + \tau_c \cdot \dot{\alpha}_c,$$

where subscript m stands for the main motor and subscript c indicated the configuration motor.

Fig. 4 shows cost of transport ($CoT = E/mgd$) for the considered robot's configuration, where (a) stands for reference open-chain hopping robot, (b) stands for the quickest closed-chain hopping robot motion and (d) for the slowest. In simulation, CoT for the open-chain robot and closed-chain robot are similar. However, in the simulation, the open-chain robot uses 4 times more power at touchdown than the closed-chain robot; in case of usage of accurate motor model, the open-chain robot could not run and "does push-ups" in place.

IV. Conclusions & Discussion

Within this paper we explore the issue of computational synthesis of a robot for dynamic interaction with an environment. The designed hopper is an illustrative example of a robot capable of dynamic locomotion driven by morphological computation. The currently available GD software focuses on generating parts shapes and their topology optimization, meanwhile we treat GD as a tool to automatically generate mechanisms to transform motion of actuators into complex trajectories of contact points satisfying a weakly formalized task of a software user. The proposed method outline for robots' co-design is a backbone for generative design (GD) of robots as systems, especially regarding the grammar rules.

Generated mechanisms should be physically reproducible, and their implementation should be justified by superior characteristics compared to previously synthesized mechanisms. The superiority of characteristics can be ensured by the morphological computation of the control laws. By mechanism superiority we mean reduced requirements for actuators' performance and sensors' accuracy, minimization of required drives, simplification of control systems, and minimization of dimensions, weight and total cost.

REFERENCES

[1] N. Roy, I. Posner, T. Barfoot, P. Beaudoin, Y. Bengio, J. Bohg, O. Brock, I. Depatie, D. Fox, D. Koditschek *et al.*, "From machine learning to robotics: Challenges and opportunities for embodied intelligence," *arXiv preprint arXiv:2110.15245*, 2021.

[2] V. C. Müller and M. Hoffmann, "What is morphological computation? on how the body contributes to cognition and control," *Artificial life*, vol. 23, no. 1, pp. 1–24, 2017.

[3] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

[4] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[5] N. Kau and S. Bowers, "Stanford pupper: A low-cost agile quadruped robot for benchmarking and education," *arXiv preprint arXiv:2110.00736*, 2021.

[6] S. Seok, A. Wang, D. Otten, and S. Kim, "Actuator design for high force proprioceptive control in fast legged locomotion," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1970–1975.

[7] G. Bledt and S. Kim, "Extracting legged locomotion heuristics with regularized predictive control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 406–412.

[8] F. Chen and M. Y. Wang, "Design optimization of soft robots: A review of the state of the art," *IEEE Robotics & Automation Magazine*, vol. 27, no. 4, pp. 27–43, 2020.

[9] J. Pinskier and D. Howard, "From bioinspiration to computer generation: Developments in autonomous soft robot design," *Advanced Intelligent Systems*, vol. 4, no. 1, p. 2100086, 2022.

[10] C. Laschi and B. Mazzolai, "Lessons from animals and plants: The symbiosis of morphological computation and soft robotics," *IEEE Robotics & Automation Magazine*, vol. 23, no. 3, pp. 107–114, 2016.

[11] G. A. Folkertsma and S. Stramigioli, "Energy in robotics," *Found. Trends Robot*, vol. 6, no. 3, p. 140–210, Oct. 2017. [Online]. Available: https://doi.org/10.1561/2300000038

[12] D. Howard, A. E. Eiben, D. F. Kennedy, J.-B. Mouret, P. Valencia, and D. Winkler, "Evolving embodied intelligence from materials to machines," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 12–19, 2019.

[13] G. A. Folkertsma, A. J. van der Schaft, and S. Stramigioli, "Morphological computation in a fast-running quadruped with elastic spine," *IFAC-PapersOnLine*, vol. 48, no. 13, pp. 170–175, 2015.

[14] P. Krcah, "Evolving virtual creatures revisited." in *GECCO*, vol. 7. Citeseer, 2007, pp. 341–341.

[15] T. Geijtenbeek, M. Van De Panne, and A. F. Van Der Stappen, "Flexible muscle-based locomotion for bipedal creatures," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 1–11, 2013.

[16] S. Ha, S. Coros, A. Alspach, J. M. Bern, J. Kim, and K. Yamane, "Computational design of robotic devices from high-level motion specifications," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1240–1251, 2018.

[17] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.

[18] T. Wang, Y. Zhou, S. Fidler, and J. Ba, "Neural graph evolution: Towards efficient automatic robot design," *arXiv preprint arXiv:1906.05370*, 2019.

[19] A. Zhao, J. Xu, M. Konaković-Luković, J. Hughes, A. Spielberg, D. Rus, and W. Matusik, "Robogrammar: graph grammar for terrain-optimized robot design," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–16, 2020.

[20] S. Chand and D. Howard, "Multi-level evolution for robotic design," *Frontiers in Robotics and AI*, vol. 8, p. 684304, 2021.

[21] I. I. Borisov, I. A. Kulagin, A. E. Larkina, A. A. Egorov, S. A. Kolyubin, and S. Stramigioli, "Study on elastic elements allocation for energy-efficient robotic cheetah leg," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1696–1701.

[22] I. I. Borisov, E. E. Khomutov, S. A. Kolyubin, and S. Stramigioli, "Computational design of reconfigurable underactuated linkages for adaptive grippers," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6117–6123.

[23] I. I. Borisov, E. E. Khomutov, D. V. Ivolga, N. A. Molchanov, I. A. Maksimov, and S. A. Kolyubin, "Reconfigurable underactuated adaptive gripper designed by morphological computation," in *2022 IEEE/RSJ International Conference on Robotics (ICRA)*. IEEE.

[24] O. V. Borisova, I. I. Borisov, K. A. Nuzhdin, A. M. Ledykov, and S. A. Kolyubin, "Computational design of closed-chain linkages: Synthesis of ergonomic spine support module of exosuit," *Nonlinearity, Information and Robotics*, 2022.

[25] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross, "Computational design of linkage-based characters," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–9, 2014.