

Deep Coverage: Motion Synthesis in the Data-Driven Era

David A. Surovik and Kostas E. Bekris

Abstract Effective robotic systems must be able to produce desired motion in a sufficiently broad variety of robot states and environmental contexts. Classic control and planning methods achieve such *coverage* through the synthesis of model-based components. New applications and platforms, such as soft robots, present novel challenges, ranging from richer dynamical behaviors to increasingly unstructured environments. In these setups, derived models frequently fail to express important real-world subtleties. An increasingly popular approach to deal with this issue is the use of end-to-end machine learning architectures, which adapt to such complexities through a data-driven process. Unfortunately, however, data are not always available for all regions of the operational space, which complicates the extensibility of these solutions. In light of these issues, this paper proposes a reconciliation of classic motion synthesis with modern data-driven tools towards the objective of “*deep coverage*”. This notion utilizes the concept of composability, a feature of traditional control and planning methods, over data-derived “*motion elements*”, towards generalizable and scalable solutions that adapt to real-world experience.

1 Modern Robotic Capabilities and Challenges

Robots have thrived in highly structured, accurately known, and safely enclosed industrial settings, such as those shown in Fig. 1 (left). Typically, these platforms consist of rigid elements connected by a few degrees of freedom (DOF) that can be controlled directly. This allows strict enforcement of motion for high speed, accuracy, and consistency of repeated tasks. In contrast, modern robotics aspires to more general and ambitious operational goals involving highly unstructured environments, arbitrary terrains, unknown objects, or the company of humans. This means that traditional objectives are superseded by a fundamental need for adaptability.

These altered priorities motivate the use of robots with versatile physical traits, such as deformation and heightened articulation, which can be achieved by using soft components or a high number of compliant DOF , as in Fig. 1 (center/right). These features allow for adaptive contact geometry and storage or dissipation of energy for purposes such as efficient mobility on rough terrain and manipulation of arbitrary objects. Examples include soft [7] and adaptive hands [8], bio-inspired modular robots [18], underactuated legged systems [27], as well as tensegrity rovers, dynamic truss structures that incorporate length-actuated members [22].

David A. Surovik and Kostas Bekris are with the Computer Science Department of Rutgers, the State University of New Jersey, USA, e-mail: {david.surovik,kostas.bekris}@cs.rutgers.edu.



Fig. 1 Left: Rigid industrial robots for manufacturing [KUKA]. Center: a soft robotic hand conforming to a grasped object [7]. Right: NASA’s terrain-compliant tensegrity rover, SUPERball [22].

While such platforms provide increased safety and adaptability, they pose many complications to the prediction and regulation of motion. Due to higher system dimensionality and imperfect sensing of environmental variation, natural dynamical influences such as momentum and contact forces become both more significant and more complex to describe. Nevertheless, a pivotal requirement is to achieve *motion coverage* with such robots. This expresses the need to effectively command motion for a wide range of operational bounds. This work aims to outline a pathway toward this goal in light of ongoing progress in motion synthesis and data-driven design.

2 Computing Motion for Operational Coverage

Discussion of the concept of motion coverage will start from *nominal* coverage, which identifies useful paths within a region, but quickly proceed to focus on *robust* coverage, which ensures intended motion under a range of circumstances.

Model-Based Motion Coverage: Given geometric and kinematic models, nominal coverage can be provided via tools for validating configurations (e.g., collision checkers [17]) and for charting connectivity, such as sampling-based roadmaps [13]. These principles can also account for dynamics [16], uncertainty [3], changing environments [26] and robustness, if given stabilizing controllers [25].

Existing motion coverage tools depend on a tractable system model.

Many systems, however, cannot easily return to a nominal path after significant deviation. *Control synthesis*, the composition of multiple control laws established prior to execution, can provide robust coverage. One composition method is to design controllers so that each region of attraction (R_{OA}) is “funneled” into another [5], as in Fig. 2 (left). LQR-trees integrate this idea with sampling to provide nonlinear robust coverage via linear quadratic regulators (LQR) [24]. Alternatively, Explicit Model Predictive Control tiles the operational space with light-weight localized equivalents of an expensive predictive controller [2] (center of Fig. 2). Composition can also address discontinuities, as in Sliding Mode Control [10].

Combining multiple controllers boosts robustness with minimal online effort.

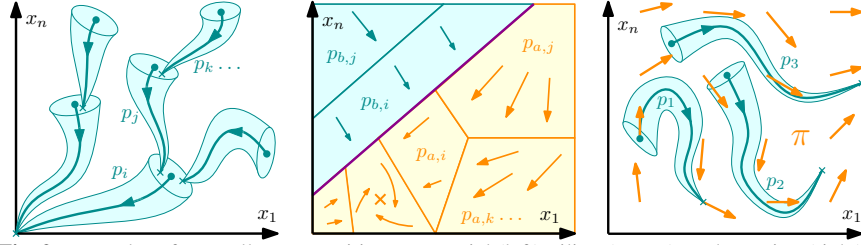


Fig. 2 Examples of controller composition: sequential (left), tiling (center), and merging (right).

Data-Driven Control: The above methods are not easily applied to systems with many compliant DoF and strong natural dynamics. Even with a descriptive model, it may be challenging to determine if a configuration of a deformable robot is valid. Contact forces, energy, and momentum hinder the identification of good dynamical paths and the derivation of stabilizing controllers. Such complications have spurred data-driven approaches for interpreting sensor data, predicting state evolution, or imposing actuation [11, 23]. They have also led into modeling all three as a single “end-to-end” process [15]. These methods fit generic parametrized models, such as polynomial functions, Gaussian processes, or artificial neural networks (ANN), to match experimental data that represent the intended operational regime.

Learned models can adequately represent complex behaviors, but typically require significant training effort for each application scenario.

The risks and demands of obtaining necessary data have motivated many innovations with relevance to controller synthesis. To improve sample efficiency via optimal control principles, Guided Policy Search merges an evolving set of LQR controllers into one ANN feedback policy [15]. Although this approach has been applied end-to-end, it can also be viewed as a controller composition scheme (see right of Fig. 2). Meanwhile, other data-driven techniques have been developed for determining nonlinear controller RoAs in the context of partially modeled dynamics and safety-constrained sampling [1, 4]. A model-free variant of LQR-Trees similarly establishes RoAs via sampling to validate the synthesized control set [20].

Recent work has begun to reveal how data-driven control can be made more efficient and verifiable, potentially for use in composition schemes.

3 Data-Driven Synthesis for Deep Coverage

An emerging possibility for motion computation on modern robotic platforms is to a) establish modular “motion elements” through an offline data-driven process and b) employ them online in an architecture that exploits their suitability for composition and synthesis. This framework, summarized in Fig. 3, will be referred to as “*deep coverage*” to emphasize the goal of effectiveness under highly variable circumstances (hence coverage) through hierarchical, modular processes that enforce robust behavior at all scopes of the problem (hence depth). Motion elements at various levels of the proposed hierarchy - which are no longer clearly delineated by an explicit, analytical model but are instead data-driven - can also be seen as equivalent to different layers in deep learning architectures for perception challenges.

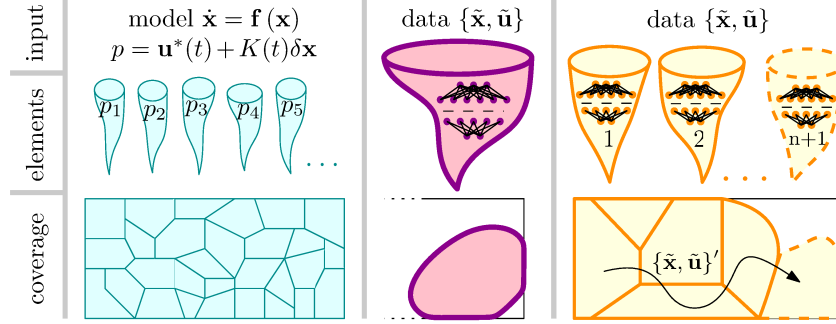


Fig. 3 Three different paradigms for computing robust motion for nonlinear, nonconvex systems. Given computational resources, motion elements are produced to provide motion coverage. Left column: model-based synthesis, center: end-to-end, right: deep coverage.

Successful employment of “deep coverage” will require careful consideration of numerous trade-offs and design principles that follow from the combined use of data-driven design and motion composition. The following discussion of some of these key aspects will frequently refer to an example scenario of deploying NASA’s tensegrity rover over an unstructured landscape, as shown in Fig. 4.

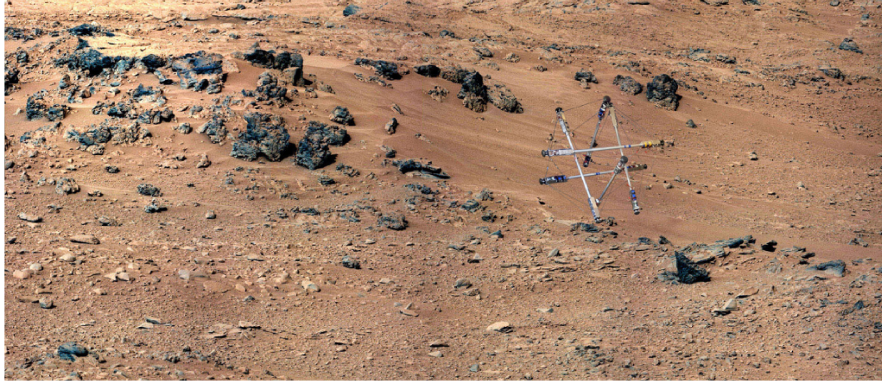


Fig. 4 A mock-up of NASA’s tensegrity rover on a rugged Martian landscape [NASA/JPL Caltech]

Robust Motion Elements: The motion elements for deep coverage must meet needs at varying physical and temporal scope. For instance, they should be able to govern the pressure response of a finger or leg, reach for or in-hand manipulate an object, or achieve locomotion of an entire platform. One way to provide this is to increase the applicability of individual elements so as to cover broader circumstances, represented as increased funnel width in Fig. 3 (center). The recent methods for verifying and expanding RoAs may help fulfill this desire through increased use of data. Training controllers to match commanded values for steering, dampening, etc., would further boost their individual usefulness.

Nonetheless, the approach of broadening the RoA through increased data use will eventually suffer diminishing returns and weakened verifiability if taken to its extreme. A contrasting desire is for each element to be *modular*, with a fundamen-

tal, distinct and well interpretable purpose. This eases the matching of elements to specific roles in different problem instances, improving adaptability [6]. If complete solutions are available as reference, modularity might be achieved by decomposing them into elements of the desired scope [12].

In the context of the tensegrity rover example, these two possibilities can be illustrated in terms of different terrain types to be defined, such as sandy or rubble-filled areas with different slopes. As depicted by the region-expanding arrows in Fig. 5, a motion element for one terrain class could potentially be broadened to cover related domains. Yet, natural limits to this practice are to be anticipated; it may be infeasible or unreasonable to formulate a single element for use, e.g., on both rubble and sand, due to fundamental differences in contact mechanics. Similar considerations could arise for a compliant end-effector, with individual elements being effective for wide varieties of grasped object shapes but naturally delineated in terms of other factors, such as object fragility or deformability. Limitations such as this would need to be recognized within reference data that span multiple types of scenarios, to give rise to appropriate boundaries of specialization.

Each motion element should be broad enough to quickly cover the space, but focused enough so that it is verifiable, modular and reusable.

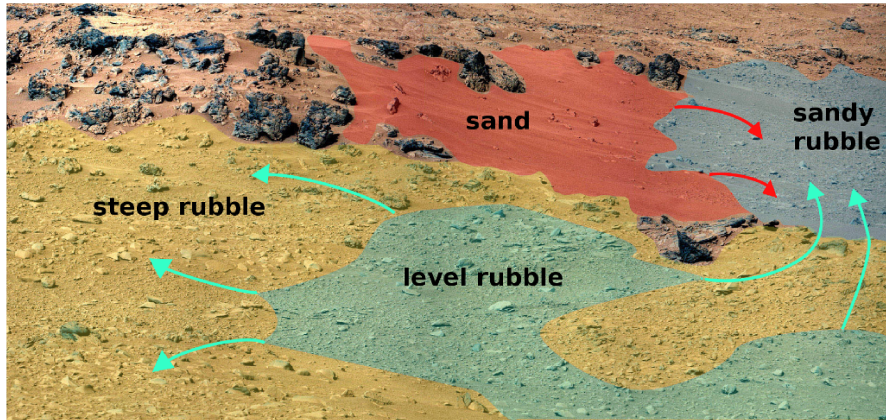


Fig. 5 Classifications of different terrains that may require different motion elements. Arrows indicate expansion of an individual element’s coverage.

Comprehensive Libraries: Equally important to the effectiveness of individual elements is that they collectively cover the underlying operational space and enable good solutions to be discovered efficiently. Covering the problem space, illustrated via tiling in the bottom of Fig. 3, requires that each class of environmental feature (e.g., terrain type in locomotion or object type in manipulation) can be handled by at least one element. Allowing overlapping applicability among elements can be beneficial for safety and performance, since the failure of one controller may be recoverable by a neighboring one. Such redundancy can also bode well for adaptability to new domains. At the same time, an over-abundance of options should be avoided in order to moderate the size of the search and decision-making process.

Approaches for designing a good *library* can borrow ideas from the principle of cooperative coevolution [19], where each individual element’s fitness is measured in the context of group performance. The data-efficiency of library design might also be improved by selectively sharing information between the training processes for different elements, possibly based on a learned classification [21].

In terms of the tensegrity rover example, this aspect suggests a requirement for both rubble-traversing and sand-traversing gaits. Less obviously, it also motivates the previously discussed broadening of individual elements — if each of these gaits can feasibly handle an intermediate terrain type containing a mixture of rubble and sand, then it may or may not be warranted to further inflate the motion element library by including an element specifically tailored to such a scenario. This is illustrated by the arrows in the “sandy rubble” region of Fig. 5, implying mutual coverage by the sand gait and the rubble gait.

An element library should be minimal but comprehensive so as to cover the operational space, while achieving efficient and robust solutions.

Composition of Elements: When efficiently piecing together a complete solution, the various interactions between motion elements, objectives, and environmental features must be considered. The traversal scenario of Fig. 6, for example, involves both different types of terrain and different demands for movement precision. For the relatively open expanse of rubble, it should quickly be recognized by the motion planner that a general-purpose low-precision gait is sufficient for progressing. Only once the cluster of sharp rocks has been reached will it become worthwhile to consider the use of more precise and specialized maneuvers, which are necessary for handling more formidable obstacles but are slower and more difficult to sequence. Other types of systems, conversely, may involve independent motion elements, such as an arm movement primitive and an end-effector grasping action. Given the holistic nature of the dynamics for compliant and deformable systems, integration of behaviors used in parallel must also be handled with care.

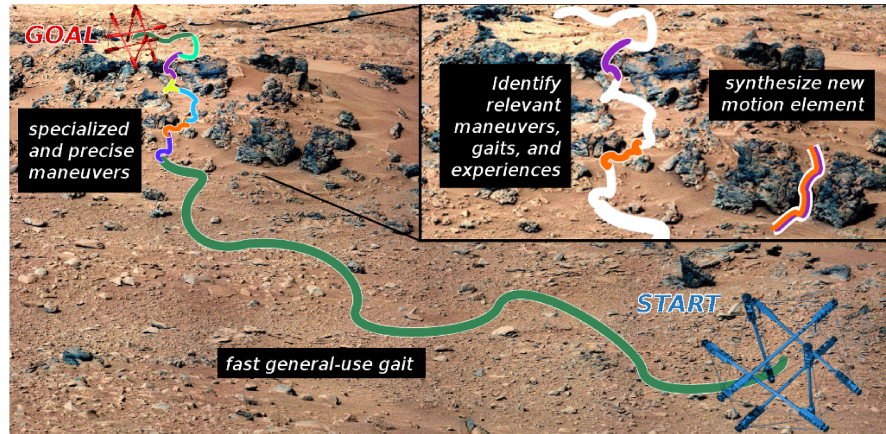


Fig. 6 A terrain navigation plan using motion elements with different scopes and properties. Call-out: Synthesis of a new motion element to address an unforeseen terrain feature.

This process could amount to an informed search of the space of element compositions, as in traditional planning, with reasoning based upon ROAs and other motion element characterizations. Alternatively, decisions could be governed by a data-driven process, aimed to handle subtle interactions. Composability of human motions, for example, can be established through segmentation of reference solutions [14]. This use of a data-driven approach for composition can lead to hierarchical schemes, in which intermediate-level elements can be responsible for the composition of low-level ones, without rigid and restrictive rules.

The motion element composition process should account for the elements' applicability and compatibility, whether determined via models or data.

Continued Learning Online: Real-world deployment of robots in natural and human-occupied workspaces inevitably leads into unforeseen setups, where a valid solution must be improvised quickly. This entails identifying a gap in coverage, safely exploring the corresponding region of the system's dynamics, and incorporating the results into a new motion element (far right of Fig. 3). These steps should benefit from offline training and effectively utilize the existing library of motion elements so that the online adaptation can be light-weight.

A benefit of data-driven approaches is that they can incorporate additional, highly-relevant data gathered in the field during deployment. Again, the challenge lies in the amount of data that need to be collected. Model-based reinforcement learning brings the hope of adaptation to new domains in a sample-efficient manner as long as the model of the robot's dynamics are appropriately adapted on the fly [9]. This type of adaptation would be most effective with a highly modular library, where elements represent the system dynamics in a principled manner.

The call-out in Fig. 6 demonstrates these principles for the tensegrity example. Traversal of the rock field would provide real-world data on the performance of some of the precise motion elements. When facing an unforeseen obstacle, such as the narrow gap between two large boulders, this information could be leveraged to guide discovery of a new motion element to fill the gap in operational coverage.

New elements should be generated on the fly to deal with unforeseen setups.

4 Discussion

Many modern robots experience complex influences from the environment and their natural dynamics, which are difficult to adequately model. This motivates end-to-end control architectures that solve motion problems at sufficient *depth*, but depend on a lot of data for generalization. Meanwhile, controller synthesis and composition enable greater *breadth* of motion coverage, but depend on accurate models. A successful reconciliation of these two paradigms can enable *deep coverage*: the efficient use of data-derived motion elements for generating robust solutions in general problem spaces in the context of exciting, frontier applications of robotics.

Acknowledgements: This work was partially sponsored by NSF IIS-1734492 and IIS-1723869.

References

1. Akametalu, A.K., Fisac, J.F., Gillula, J.H., Kaynama, S., Zeilinger, M.N., Tomlin, C.J.: Reachability-based safe learning with Gaussian processes. In: CDC, pp. 1424–1431 (2014)
2. Alessio, A., Bemporad, A.: A Survey on Explicit Model Predictive Control (2009)
3. Bai, H., Hsu, D., Lee, W.S.: Integrated perception and planning in the continuous space: A POMDP approach. *IJRR* **33**(9), 1288–1302 (2014)
4. Berkenkamp, F., Moriconi, R., Schoellig, A.P., Krause, A.: Safe learning of regions of attraction for uncertain systems with Gaussian processes. In: CDC, pp. 4661–4666 (2016)
5. Burridge, R.R., Rizzi, A.A., Koditschek, D.E.: Sequential Composition of Dynamically Dexterous Robot Behaviors. *IJRR* **18**(6), 534–555 (1999)
6. Clune, J., Mouret, J.B., Lipson, H.: The evolutionary origins of modularity. *Proc. R. Soc. B* **280**(1755), 2012–2863 (2013)
7. Deimel, R., Brock, O.: A novel type of compliant and underactuated robotic hand for dexterous grasping. *IJRR* **35**(1-3), 161–185 (2016)
8. Dollar, A.M., Howe, R.D.: The Highly Adaptive SDM Hand: Design and Performance Evaluation. *IJRR* **29**(5), 585–597 (2010)
9. Fu, J., Levine, S., Abbeel, P.: "one-shot learning of manipulation skills with online dynamics adaptation and neural network priors". In: IROS (2016)
10. Guldner, J., Utkin, V.I.: Sliding mode control for gradient tracking and robot navigation using artificial potential fields. *IEEE TRA* **11**(2), 247–254 (1995)
11. Hou, Z.S., Wang, Z.: From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences* **235**, 3–35 (2013)
12. Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Comp.* **25**(2), 328–373 (2013)
13. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE TRA* **12**(4), 566–580 (1996)
14. Lee, C.S., Elgammal, A.: Human motion synthesis by motion manifold learning and motion primitive segmentation. In: Articulated Motion and Deformable Objects, pp. 464–473 (2006)
15. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. *JMLR* **17**(39), 1–40 (2016)
16. Li, Y., Littlefield, Z., Bekris, K.E.: Asymptotically optimal sampling-based kinodynamic planning. *IJRR* **35**(5), 528–564 (2016)
17. Lin, M., Manocha, D.: Efficient Contact Determination in Dynamic Environments. *International Journal of Computational Geometry & Applications* **07**(01n02), 123–151 (1997)
18. Onal, C.D., Rus, D.: Autonomous undulatory serpentine locomotion utilizing body dynamics of a fluidic soft robot. *Bioinspiration & Biomimetics* **8**(2), 3–26 (2013)
19. Pollack, J.B., Lipson, H., Ficici, S., Funes, P., Hornby, G., Watson, R.A.: Evolutionary Techniques in Physical Robotics. In: Evolvable Systems: From Biology to Hardware (2000)
20. Reist, P., Preiswerk, P., Tedrake, R.: Feedback-motion-planning with simulation-based LQR-trees. *IJRR* **35**(11), 1393–1416 (2016)
21. Rennie, C., Bekris, K.E.: Discovering a library of rhythmic gaits for spherical tensegrity locomotion. In: IEEE ICRA (2018)
22. Sabelhaus, A.P., Bruce, J., Caluwaerts, K., Manovi, P., Firoozi, R.F., Dobi, S., Agogino, A.M., SunSpiral, V.: System design and locomotion of SUPERball, an untethered tensegrity robot. In: IEEE ICRA, pp. 2867–2873 (2015)
23. Tai, L., Liu, M.: Deep-learning in Mobile Robotics - from Perception to Control Systems: A Survey on Why and Why not. *arXiv:1612.07139 [cs]* (2016)
24. Tedrake, R., Manchester, I.R., Tobenkin, M., Roberts, J.W.: LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification. *IJRR* **29**(8), 1038–1052 (2010)
25. Walsh, G., Tilbury, D., Sastry, S., Murray, R., Laumond, J.P.: Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Trans. Autom. Control* **39**(1), 216–222 (1994)
26. Yang, Y., Brock, O.: Elastic roadmaps—motion generation for autonomous mobile manipulation. *Autonomous Robots* **28**(1), 113 (2010)
27. Zhou, X., Bi, S.: A survey of bio-inspired compliant legged robot designs. *Bioinspiration & Biomimetics* **7**(4) (2012)