ASME Letters in Translational Robotics
Online journal at:
https://asmedigitalcollection.asme.org/letterstransrobotics

Design-Centric Robotics Learning: Deploying a Modular, Mechanism-Based Kit From Lab to K-12 and Postsecondary Education

Anurag Purwar
Department of Mechanical Engineering,
Computer-Aided Design and Innovation Lab,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: anurag.purwar@stonybrook.edu

We present SnappyXO® Design, a low-cost educational robotics platform that supports design-centric learning from K-12 through postsecondary education. The platform integrates (1) a modular hardware kit built from planar, snap-fit components, (2) companion software for mechanism and part design as well as robot control, and (3) standards-aligned instructional materials. The SnappyXO hardware uses a minimal library of planar structural elements and compliant snap-fit "alphabet" connectors that encode geometric and kinematic constraints, enabling rapid reconfiguration into a wide variety of mechanisms and robots. The software suite includes tools for linkage synthesis and simulation, custom part design for digital fabrication, and mobile control of microcontroller-based robots. Together, these elements emphasize the full engineering design cycle, from mechanism concept generation and prototyping to programming and testing, in a classroom-friendly workflow.

We motivate the need for SnappyXO by limitations of existing kits, including high cost, constrained design freedom, and weak alignment to engineering design standards. We summarize results from pilot programs and National Science Foundation (NSF) Small Business Technology Transfer (STTR) Phase I/II case studies, which indicate that learners across age groups can successfully design and build original robotic systems using SnappyXO, with positive impacts on engagement and targeted learning outcomes. SnappyXO was originally developed at Stony Brook University and advanced through multiple NSF awards. [DOI: 10.1115/1.4070994]

Keywords: STEM education, robotics, kinematic synthesis, machine learning, linkage mechanisms, translational research results from innovative designs, use-inspired innovations, risks and practical implications

1 Introduction

Robotics offers a natural context for interdisciplinary STEM learning because it couples physical construction with sensing, actuation, and computation. Through designing, building, and programming robots, learners engage with concepts from mechanics, electronics, and software while practicing iterative problem solving and teamwork. At the same time, the growing importance of robotics and embodied artificial intelligence in automation, healthcare, and service industries increases the value of early, authentic exposure to engineering design and systems thinking [1]. National reports and standards further motivate this need: the Science & Technology Action Committee emphasizes the importance of expanding and diversifying the domestic Science, Technology, Engineering, and Mathematics (STEM) workforce [2], while Next Generation Science Standards (NGSS) foregrounds engineering design as a core practice across K-12 science education.

Despite these drivers, robotics instruction in many schools remains limited in scope and accessibility. Cost and logistical constraints often restrict robotics to extracurricular clubs or competitions, rather than broad, in-class adoption. Moreover, many widely used kits emphasize assembly of predefined models and incremental programming exercises, which can limit opportunities for students to engage in open-ended mechanism and system design. Continuity across grade levels is also challenging: platforms suitable for elementary grades may not scale to high school or early college needs, leading districts to adopt multiple incompatible systems. Similar gaps appear in higher education, where early-year curricula often struggle to provide hands-on, design-oriented experiences at scale in large enrollment courses.

Manuscript received August 10, 2025; final manuscript received January 22, 2026; published online March 3, 2026. Assoc. Editor: Xiangyun Li.

In this article, we present SnappyXO® Design, a robotics learning platform developed to address these barriers. SnappyXO consists of three integrated components: (1) a modular hardware kit with planar structural parts and compliant snap-fit connectors that encode geometric and kinematic constraints, (2) software tools for mechanism design, custom part creation, and robot control, and (3) standards-aligned curriculum and instructional resources for educators. Together, these components support an end-to-end workflow in which students can generate a mechanism concept, prototype and iterate physically, and program and evaluate the resulting robot (Fig. 1).[1]

The remainder of the article is organized as follows. Section 2 reviews prior work in educational robotics and summarizes limitations of existing kits that motivated our design requirements. Section 3 details the SnappyXO hardware principles and innovations, followed by the companion software applications in Sec. 4 and pedagogical integration in Sec. 5. Section 6 presents case studies and results from National Science Foundation (NSF) STTR Phase I/II pilots, and Sec. 7 discusses comparative advantages and value propositions. Section 8 concludes with future directions.

2 Background: Educational Robotics and Challenges

A substantial body of prior research demonstrates that robotics can serve as a powerful interdisciplinary learning platform. Students engaged in robotics activities often show improved understanding of STEM concepts and increased motivation [3–5]. Robotics projects require learners to apply scientific and mathematical knowledge (e.g., geometry for mechanism design, physics for motion), practice engineering design processes (brainstorming, prototyping, testing), and develop computational skills through programming. This integration of disciplines supports the development of transferable skills such as problem solving, creativity, and collaboration [6]. Robotics has also been shown to engage a broad range of learners by providing a tangible context for applying abstract ideas, particularly when students are encouraged to design and iterate on their own solutions rather than follow prescribed instructions.

Educational robotics (ER) [7] has evolved from the use of robots primarily as motivational tools to a research-informed instructional medium spanning K-12 computing, integrated STEM education, and introductory university-level engineering and programming. In K-12 contexts, the strongest evidence base centers on computational thinking (CT) and STEM learning outcomes. Systematic reviews of ER interventions for young learners (pre-K-6) report a focus on core CT practices such as sequencing, conditionals, loops, and debugging, often implemented through collaborative and project-based pedagogies [8]. Meta-analytic evidence from primary education further indicates that ER can support STEM competence development when design and programming activities are integrated with sound instructional practices [9]. In higher education, ER is most commonly used in introductory programming, mechatronics, or robotics laboratories, where adoption studies emphasize the importance of institutional readiness, pedagogical integration, and instructor support for successful scaling [10]. Comprehensive reviews of K-12 implementations are provided by Anwar et al. [11].

Limitations of Existing Robotics Kits. Despite the documented educational benefits of robotics, many commercially available kits limit students' engagement with authentic engineering design. Popular modular platforms (e.g., LEGO® SPIKE Prime [12], VEX V5 [13], Tetrix [14], Fischertechnik [15], and Makeblock [16]) typically emphasize step-by-step assembly of predefined models followed by basic programming. While effective for introducing programming concepts and familiarizing students with mechanical components, such instruction-driven approaches rarely require learners to conceptualize original mechanisms or explore alternative structural solutions. As a result, students often replicate existing designs rather than engage in invention, which can constrain creativity and limit exposure to core engineering practices.

Cost and scalability present additional barriers to widespread adoption. Many kits are expensive and must be shared among multiple students, which often restricts robotics activities to extracurricular clubs or competitive teams. High price points reduce access for schools with limited budgets and discourage one-to-one student engagement. Furthermore, reliance on proprietary electronics and closed ecosystems can lock institutions into single vendors and inhibit the use of low-cost, off-the-shelf components that are common in educational and professional practice.

Continuity across grade levels is another persistent challenge. Platforms designed for elementary students often do not scale to the needs of secondary or postsecondary learners, forcing schools to adopt multiple, incompatible systems. At the same time, teachers frequently report limited preparation and confidence in facilitating open-ended engineering design activities within required science or technology courses. These issues contribute to a gap between the aspirations of standards such as NGSS, which emphasize engineering design, and the realities of classroom implementation. Similar gaps exist in higher education, where design-oriented robotics experiences are often introduced late in the curriculum rather than during formative early years. Many of these challenges were corroborated through customer discovery conducted as part of an NSF I-Corps project [17].

Design Goals for a New Solution. Addressing these limitations requires a robotics platform that emphasizes open-ended design, remains affordable at classroom scale, and supports a coherent learning progression from elementary through higher education. Such a platform should enable students to create and iterate on diverse mechanisms rather than assemble fixed models, while providing educators with curriculum and tools aligned to academic standards. These design goals informed the development of SnappyXO Design, which is described in the following sections.

3 SnappyXO Hardware Design: Principles and Innovations

The SnappyXO hardware kit was designed from the ground up to enable versatile, design-centric prototyping while being low-cost and classroom-friendly. The kit consists of a set of planar *only* plastic parts for structural elements, mounts for off-the-shelf electronics and microcontrollers, and a variety of snap-fit connectors that join these parts in different configurations. All components are reusable and modular, allowing endless reconfiguration into different robots or mechanical contraptions. In developing the hardware, we identified several core design and manufacturing requirements:

(1) *Wide design space*: Create a minimal, unified library of planar components that can span a broad range of possible robot designs. In essence, a small number of part shapes should yield a large variety of mechanisms and structures.
(2) *Cross-grade usability*: Ensure the same set of components is usable by students from elementary school to college. The hardware should be simple enough for a 3rd grader yet capable enough for an engineering freshman, facilitating continuity in learning.
(3) *Pedagogical simplicity*: Keep the number of distinct part types to a minimum. Each part type should be general-purpose, and parts can be logically grouped (by function or difficulty) to align with pedagogical progression.

[1]https://youtu.be/ltE0TEEf4mg

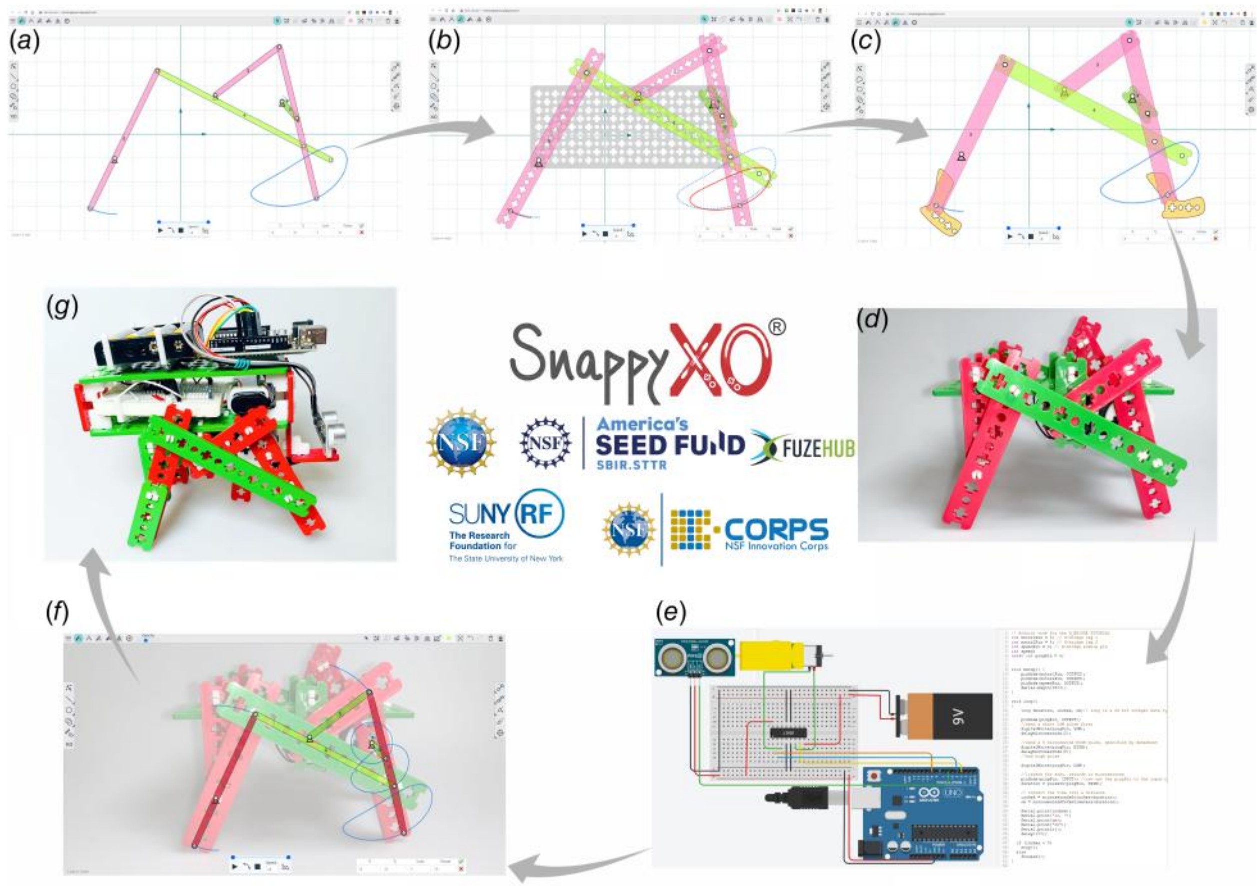

Fig. 1 Stages of design cycle for a walking robot: (*a*) users sketch a mechanism design concept (simulation mode) or input the required walking motion (synthesis mode) using MOTIONGEN software part of the SnappyXO ecosystem, (*b*) software generates mechanism design concepts for enabling walking motion and shows closest SnappyXO hardware for a chosen design, (*c*) users can also sketch custom components, such as boots and attach to the mechanism, (*d*) physically prototype the mechanism using SnappyXO hardware and optionally 3D print or laser-cut custom parts, (*e*) design and program the circuit using inexpensive, off-the-shelf electronics and open-source microcontrollers, (f) import the design back in the software for validation, and (*g*) present final design. See this walking machine driven by motors alone in footnote 1.

(4) *Safety and ergonomics*: Design components to be safe, durable, and age-appropriate. Parts are sized for easy handling, have no sharp edges, and connections can be made/re-made without tools or fasteners.
(5) *2D manufacturing*: Parts should be manufacturable using a 2D profile cutting process, such as CNC laser- or water-jet cutting to enable quick custom design and prototyping; alternatively, they can be 3D printed as well at a lower cost and time than true 3D parts. Design should permit usage of different grade of materials, such as wood, Acrylic, or Acetal.

These requirements drove the physical design of SnappyXO. Figure 2 shows a selection of the SnappyXO hardware parts, including both structural elements (beams and plates of various lengths), connector pieces of different shapes, and mounts for off-the-sheld electronics.

The structural parts are flat (2D) elements with a series of regularly spaced holes called "X" or "O." Incidentally, the X-holes look like a plus (+) sign in our parts, but are called X-holes. Each hole is a standardized interface point where connectors can attach. The decision to use strictly planar parts was deliberate: it simplifies manufacturing (all parts can be laser-cut from flat sheets of 1/8-in. thickness) and storage, and it also makes it easy for students to design and make their own parts (e.g., cutting custom shapes from sheet material), if needed. Despite being planar, these parts can be assembled into 3D structures (e.g., using perpendicular clips to join beams at right angles).

The holes in each part serve as universal connection points; they are sized to accommodate the snap-fit connectors with a slight interference for a firm grip. The connectors, often referred to as "alphabet clips" due to their letter-like shapes (L, C, H, I, etc.), are perhaps the most innovative aspect of SnappyXO's hardware; see Fig. 3 for a few of these connectors. Each connector is a small piece with prongs or jaws that snap into the holes of the beams/plates and provide either a rigid connection or rotary/sliding joint. These connectors are minimally designed to allow users to associate geometry of a part with its function intuitively and enable them to *grow* their creations along all possible directions in the 3D space. Remarkably, we found that one needs very few such pieces to be able to realize 3D Cartesian geometries.

By design, different clip shapes impart different geometric constraints when joining two or more structural pieces. For example, an L-clip holds two beams at a fixed 90-degree angle (enforcing perpendicularity), a C-clip joins two beams in the same plane (coplanarity), and an H-clip holds two parallel beams at a set spacing (enforcing parallelism). Other connectors create hinged joints: the I- and F-clips with smaller tooth geometry have at least one end freely rotating inside an "O-shaped" hole of beams/plates, effectively functioning as a revolute joint in mechanisms. Even these connectors are cut from flat sheets of plastics and as a result are not required to be axi-symmetric cylindrical pieces commonly found in other kits. By mixing and matching these connector types, students can construct complex linkages, for instance, a four-bar linkage or even multi-degree-of-freedom mechanisms,

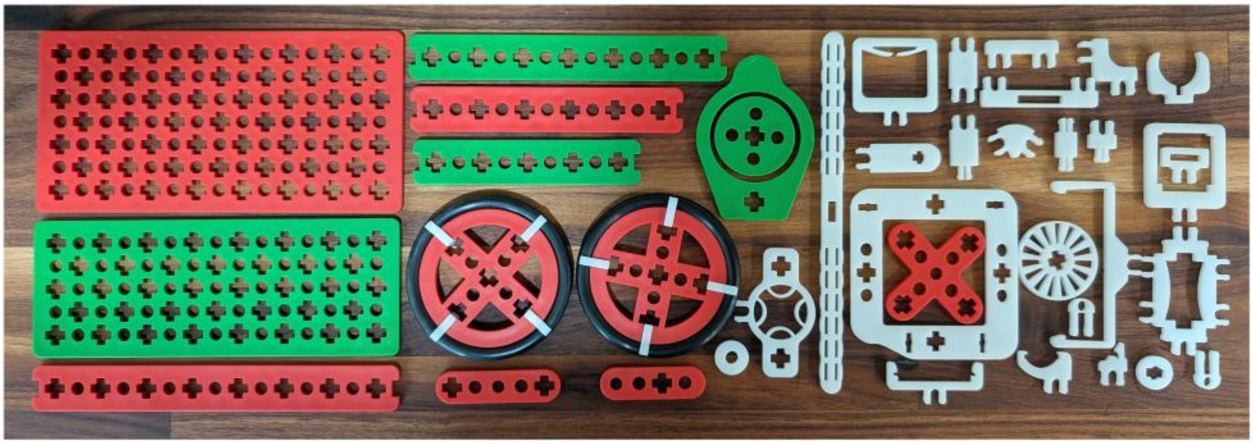

Fig. 2 A few exemplary standard structural parts, clips, and mounts for motors, sensors, etc., of the SnappyXO hardware laser-cut from nominal 1/8-in. thick Acetal Homo-polymer sheets

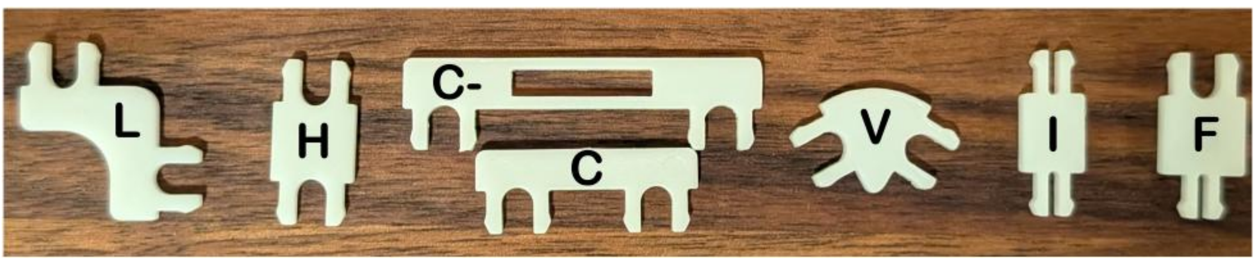

Fig. 3 Alphabet clips encode common geometric/kinematic constraints (e.g., perpendicularity, coplanarity, offset, pivots) using planar parts

using only flat beams and the appropriate clips as joints. The system thus captures the essence of mechanism design: the beams serve as links and the clips serve as joints or supports in the mechanical structure of a robot. A crucial innovation in SnappyXO is the use of compliance in the connectors to achieve robust snap-fit behavior. The connectors are made of a semiflexible Acetal polymer (commonly known as Delrin), and their geometry (such as the shape of clip "teeth") is engineered to flex just enough to snap into a hole and then hold tightly. This built-in compliance allows the same connector to accommodate slight variations in material thickness or hole size, which is important because our parts could be made from different sheet materials (plastic, wood, etc.) and because the laser-cutting process has minor dimensional tolerances. See Fig. 4 for a few example contraptions.

3.1 Manufacturing and Fit: Materials, Tolerances, and Durability. Material selection in SnappyXO directly supports repeated snap-fit engagement, long-term durability, and manufacturability across multiple fabrication methods. Polyoxymethylene (POM), commonly known as Acetal, is the primary material used for connectors and most structural components due to its favorable mechanical and tribological properties. Its elastic flexure enables compliant snap-fit behavior, while low friction, wear resistance, and dimensional stability minimize insertion forces, tooth wear, and long-term creep over repeated assembly cycles. Acrylic, although easy to laser-cut with high edge quality, is significantly more brittle and susceptible to crack initiation at stress concentrators under repeated snap loading; it is therefore recommended primarily for noncompliant structural parts. Wood introduces additional variability due to anisotropy and sensitivity to humidity; repeated insertions can compress fibers around holes and progressively reduce retention. As a result, wood is likewise best suited for rigid, nonsnap-fit components.

The tolerance strategy of SnappyXO prioritizes functional robustness over tight dimensional control by relying on elastic deformation rather than brittle interference. All critical interface features, including standardized X-holes and O-holes, are dimensioned in millimeters with deliberate positive tolerances (e.g., +0.10 mm) to avoid over-constraint while preserving reliable engagement. O-holes provide sufficient clearance for low-friction rotational motion, even under molding shrinkage or minor dimensional drift.

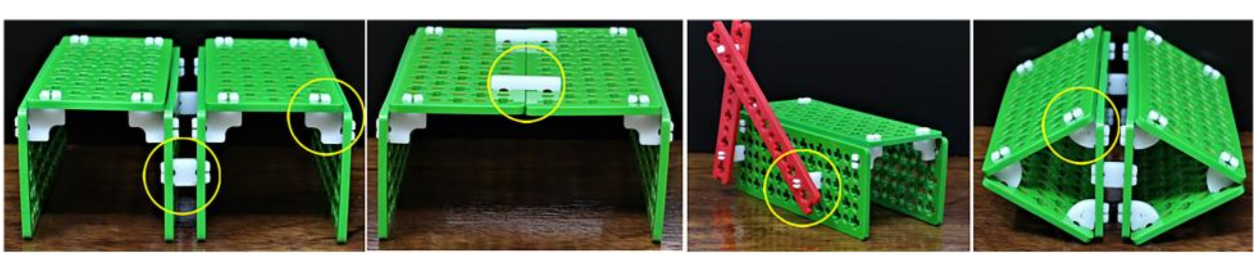

Fig. 4 A few simple SnappyXO Structures made from alphabet clips; these clips enforce different geometric constraints between structural elements, such as parallelism, incidence, angular, rotation, and sliding constraints

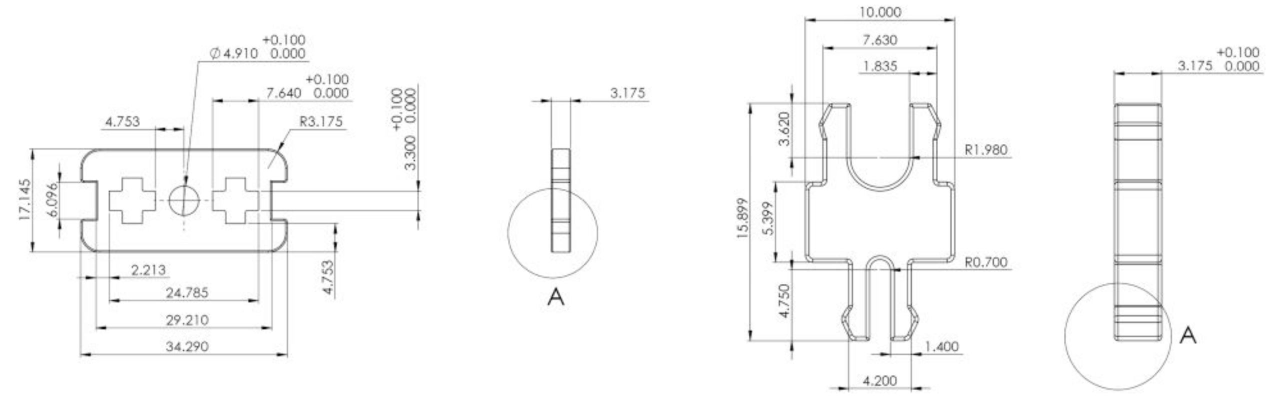

Fig. 5 Drawings of a 2X Beam (Part No. MD152) on the left and a small Fixed Pivot (Part No. MD222) on the right

X-holes employ asymmetric tolerance allocation to accommodate tooth insertion and thickness stacking while maintaining retention. Snap-fit performance is governed primarily by controlled flexure of compliant teeth rather than precise hole sizing, which reduces sensitivity to manufacturing variation. Durability is further enhanced through generous root radii (e.g., R0.7–R3.175 mm) and the elimination of sharp internal corners, mitigating stress concentrations during repeated insertion and removal cycles. Figure 5 shows example drawings of a 2X Beam and a fixed pivot.

SnappyXO components are manufactured using both laser cutting and injection molding, and the hardware design explicitly accommodates the characteristics of each process. Early prototypes and low-volume parts were laser-cut from sheet Acetal, enabling rapid iteration with minimal tooling cost. Laser cutting introduces kerf taper and greater part-to-part variability, which are mitigated through compliant connector geometry and tolerance design. Injection molding, used for all snap-fit connectors and most standard structural components, provides significantly improved dimensional consistency, eliminates taper, and allows tighter control of material properties. In effect, injection molding functions as a highly repeatable fabrication process for these parts, while preserving compatibility with laser-cut components. This hybrid manufacturing strategy enables low-cost prototyping, scalable production, and consistent user experience across fabrication methods, with snap-fit reliability governed primarily by material elasticity rather than process precision.

In summary, the patent-pending hardware design [18] specifically emphasizes (1) a novel way to enforce structural and kinematic constraints using only planar pieces, and (2) the strategic use of compliance in snap-fit connectors to enable reliable connections across various materials and thicknesses. These features differentiate SnappyXO from rigid, bolted systems and make it particularly well suited for educational use, where connections are secure but can be undone without special tools, enabling quick design iterations. Figure 6 shows a few examples of linkage-based robots.

3.2 Integration With Off-the-Shelf Electronics and Custom Design. SnappyXO hardware is designed to seamlessly integrate common off-the-shelf electronics, instead of relying on proprietary parts. In fact, this has been a persistent challenge in DIY robotics where users have needed a consistent and reliable method for mounting off-the-shelf electronics. The kit includes mounts and holders for components like Arduino® Uno microcontroller boards, DC motors and servo motors, standard hobby sensors (ultrasonic rangefinders, IR sensors, encoders, etc.), battery packs, and breadboards. Each of these mounts uses the same hole interface, meaning they connect with structural parts using provided clips. For example, an "Arduino mount" is a laser-cut plate that the Arduino snaps on; that plate has holes for clips so the microcontroller can be attached to the robot chassis. By using widely available electronics and open-source platforms (Arduino, Micro:Bit, Raspberry Pi, etc.), SnappyXO dramatically lowers the cost of a complete robot system and gives students exposure to industry-standard tools. In contrast, many commercial kits use custom electronics or a closed programming environment limited to that kit.

With SnappyXO, a student can program their robot in Arduino C/C++, block-based, or any language supported by the microcontroller of choice, and they can interface standard sensors and actuators, thereby learning real-world skills in electronics and coding. The open architecture also means the platform keeps pace with technology: as new microcontrollers or sensors come out, they can be adopted easily because the system is not locked to a specific proprietary ecosystem. Moreover, we also provide an intuitive and simple 2D CAD software (discussed later) for students to also design and fabricate highly customized parts and mounts for their own electronics; see Fig. 7 for example. This flexibility is highly valued by educators who want longevity from their purchases and compatibility with existing school makerspace equipment.

Affordability is a key tenet of SnappyXO's design. By using flat components, we leveraged low-cost manufacturing techniques like

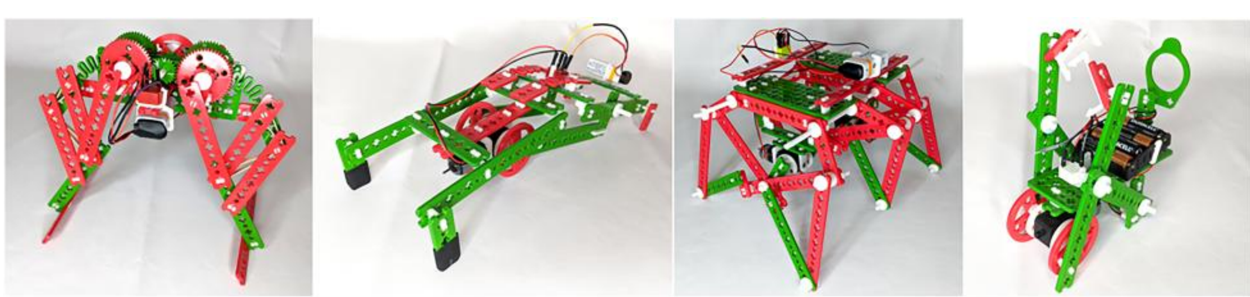

Fig. 6 A few examples of linkage-based robots showing SnappyXO parts, such as beams, plates, wheels, gears, connectors, and mounts for electronics; most parts have a pattern of X- and O-holes cut in them for inserting structural and motion connectors, respectively

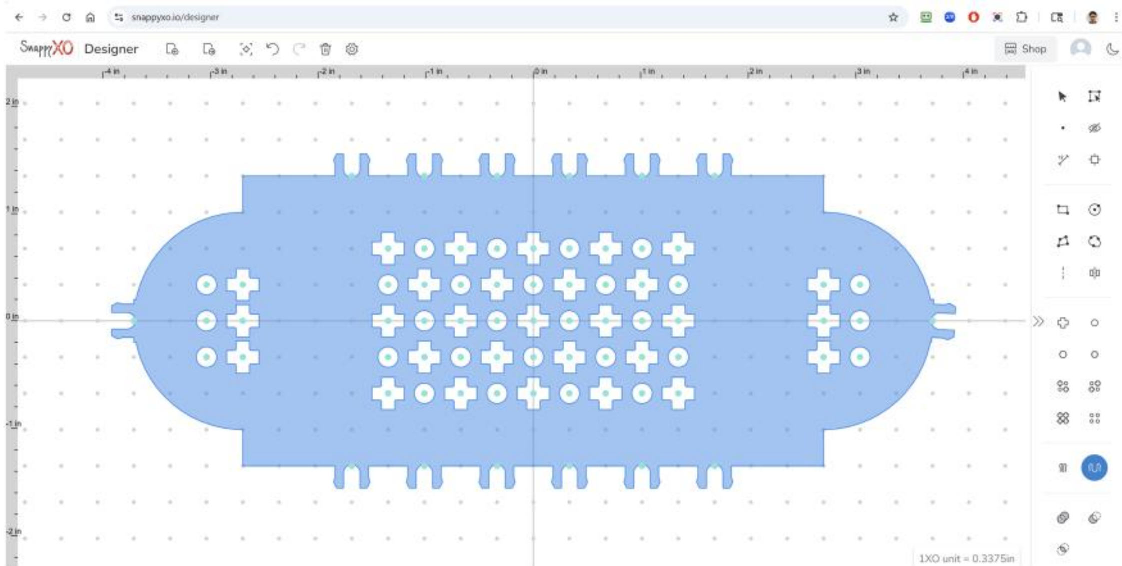

Fig. 7 A screenshot of the DESIGNER application showing a custom chassis with internal X and O holes using SnappyXO pitch and external teeth geometry for attaching other parts

CNC laser-cutting for initial development and production. Initially, all standard SnappyXO parts were cut from sheets of Acetal plastic sheets of 1/8 inch thickness. Laser-cutting required minimal upfront tooling cost and allows rapid design revisions, which was invaluable during research and development. It also makes it feasible for schools with laser cutters to fabricate their own additional parts or custom designs. Later on, injection molding for mass production reduced per-unit cost and allowed more precise control of part dimensions and material properties.

Resulting Limitations. Since the primary kinematic expressivity is planar (e.g., linkages with joint axes normal to the part planes), spatial mechanisms with nonparallel joint axes (e.g., universal/spherical joints, bevel gears, spatial linkages) generally require additional 3D adapter parts or a different part architecture. Moreover, out-of-plane stiffness and torsional rigidity are typically lower than beam-based 3D construction systems; achieving high rigidity often requires stacking plates, triangulation, or additional bracing, increasing part count.

Trade-Offs Versus True 3D Parts. Planar parts reduce SKU complexity, and enable low-cost scalable fabrication, easier custom part creation (2D CAD laser-cut), fast iteration, and compatibility with mixed sheet materials. But, true 3D geometries have fewer parts per structure, higher torsional stiffness, integrated features, and simpler realization of spatial joints.

Due to the simpler geometries and use of commodity electronics, the cost of SnappyXO hardware is lower than other comparable kits. This cost advantage translates directly to educational impact: schools can purchase more kits with the same budget, allowing more students to participate simultaneously, and even permitting students to take kits home for extended learning. In summary, the SnappyXO hardware is engineered for maximum versatility and educational value per dollar, prioritizing the needs of students and teachers in real classroom settings.

4 Software Applications

In this section, we review four software applications specifically developed to facilitate the process of designing, building, and controlling robotic systems. MOTIONGEN enables students to simulate and synthesize mechanical linkages, facilitating a deeper understanding of mechanism synthesis for robotic applications. DESIGNER supports the creation of custom components tailored for integration with our standard hardware, thereby promoting design adaptability and flexibility. MICROBLUE provides a platform for wireless control of robots and peripheral electronic systems via low-energy Bluetooth (BLE). Finally, PROJECTS offers a structured learning environment where students can engage with exemplar robotics projects of varying complexity, enabling them to acquire hands-on experience in design, construction, and programming within an integrated framework. All of these applications and curriculum are available freely in online.[2]

4.1 MOTIONGEN: Artificial Intelligence-Powered Mechanism Design Platform.

MOTIONGEN[3] is a cloud-based, AI-driven platform that automates the synthesis and simulation of mechanical linkages and robotic mechanisms directly from motion intent. Built on over a decade of research in kinematics, machine learning, and computational design [19–48], MOTIONGEN addresses a critical gap in current engineering workflows: early-stage motion concept generation. Traditional CAD tools require manual topology selection, iterative geometric modeling, and simulation expertise. MOTIONGEN disrupts this process by leveraging neural networks and curated motion-mechanism datasets to learn the mapping from desired motion trajectories to feasible mechanism topologies and dimensions. At its core, the platform incorporates a motion-to-mechanism (M2M) engine that encodes mechanism structures. Using generative models (e.g., variational autoencoders and its variants), MOTIONGEN proposes candidate designs for a given motion that are physically realizable. Figure 8 shows a few example mechanisms modeled in MOTIONGEN. In undergraduate Kinematic of Machinery class [49,50], students have successfully used MOTIONGEN to design and simulate planar linkage mechanisms.

4.2 DESIGNER: A 2D CAD Application for Designing Custom Robot Parts.

DESIGNER is also a web-based application,[4] which enables students to sketch their own part designs intuitively in a 2D CAD interface and export for 3D printing and laser-cutting. Users can use draw simple shapes (like rectangle, circle) as well

[2]http://www.snappyxo.io
[3]http://www.motiongen.io
[4]http://www.snappyxo.io/designer

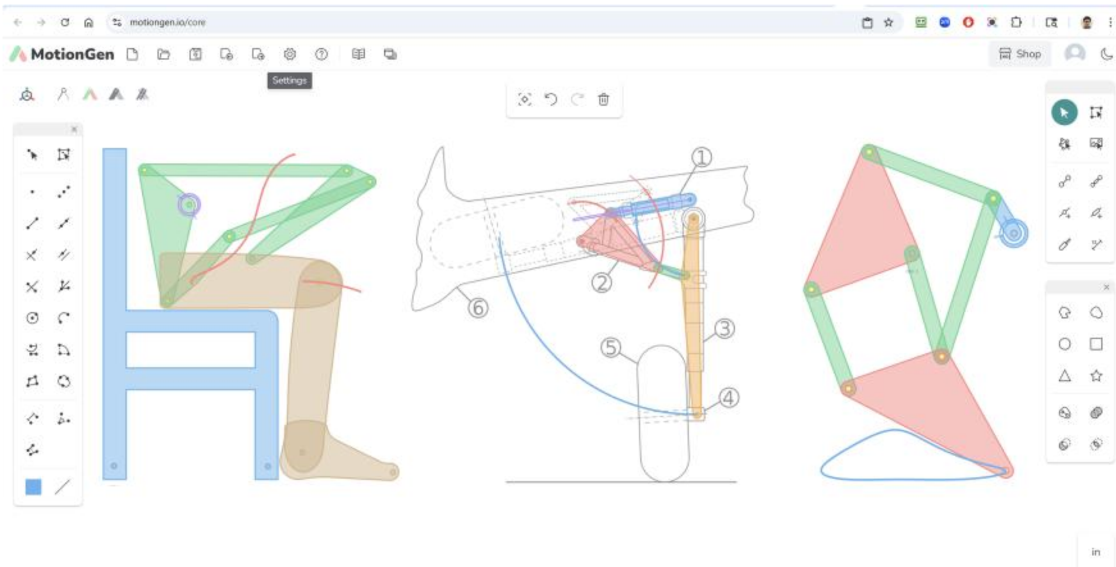

Fig. 8 A screenshot of the MOTIONGEN application showing three different simulated mechanisms; from left to right: for a sit-to-stand device, aircraft landing gear, and Jansen's walker

as closed free-form ones and perform regularized boolean operations on them to generate new shapes. The X and O holes as well as hole features for DC and servo motor shaft can be dropped on them to create parts that are compatible with the standard hardware and electronics.

Figure 7 shows a custom chassis with alternating pattern of X and O holes in the middle with rows of X- and O-holes only at the two ends. This shape was created by performing a union operation on a rectangle and two circles at the end and thereafter holes were simply dropped. On the boundaries of the shape, one can see the teeth features added to enable directly connecting beams, plates and other parts. Figure 9 shows a few custom robots with user-designed parts.

4.3 MICROBLUE: An iOS and Android Application for Controlling Robot Motions.

MICROBLUE is a mobile application designed to provide a flexible, low-latency wireless control interface for microcontroller-based robotic and mechatronic systems, with particular emphasis on educational and research environments. The application establishes secure connections over Bluetooth Low Energy (BLE), utilizing the Generic Attribute Profile (GATT) for structured data exchange and supporting custom writable services to ensure reliable, interference-free communication with compatible hardware platforms. These platforms include Arduino microcontrollers (e.g., Arduino Uno paired with HM-10 BLE modules), ESP32-series devices, and Raspberry Pi Pico(W) boards, enabling seamless integration across a wide range of robotic architectures.

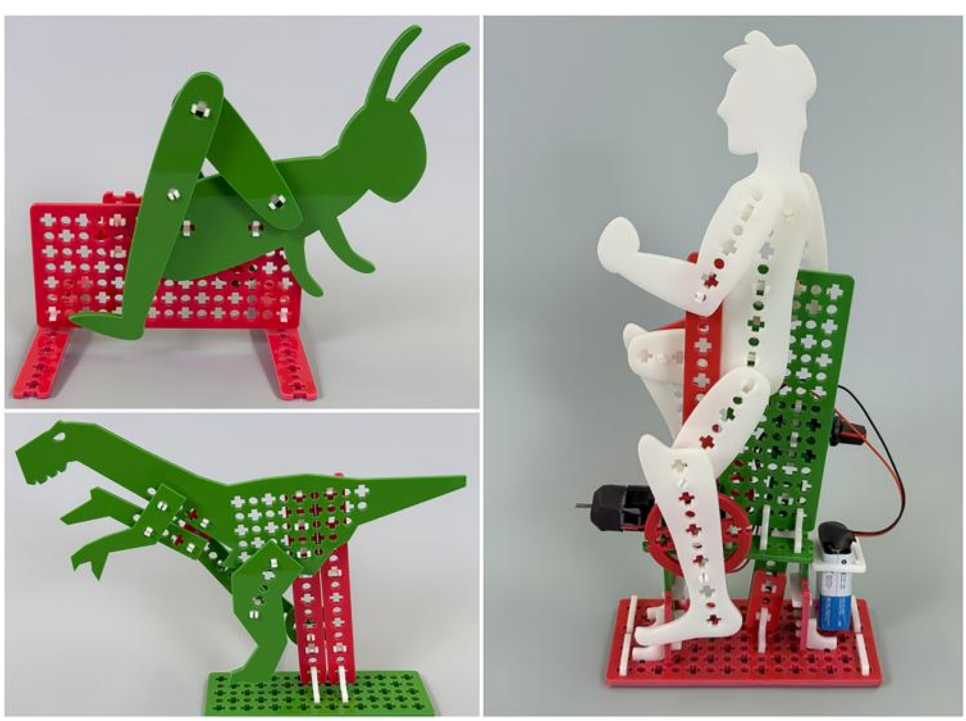

Fig. 9 Custom robots showing user-designed parts connected to standard hardware

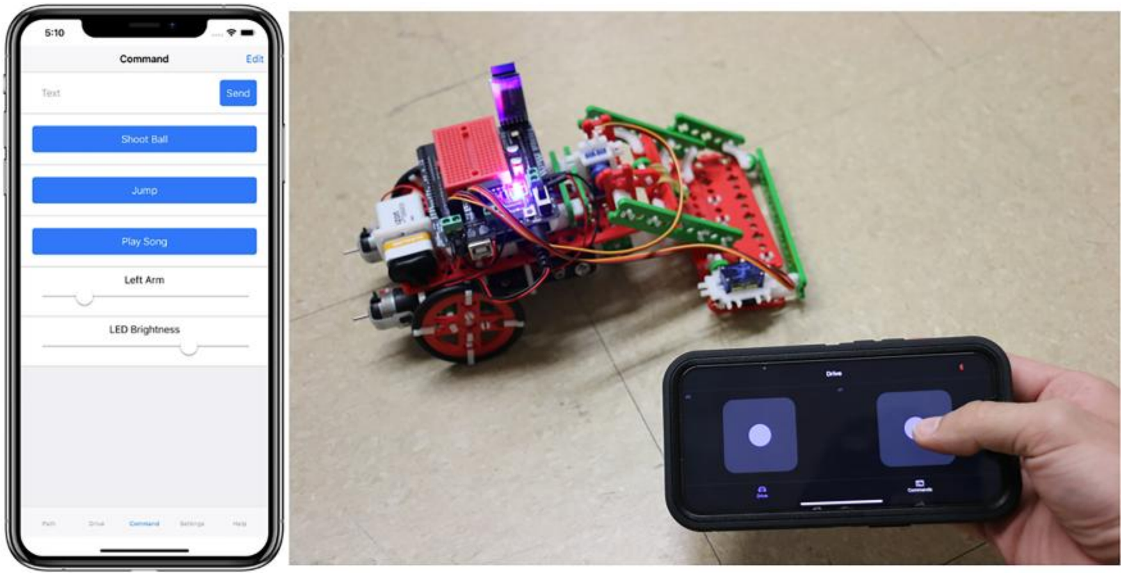

Fig. 10 Left: interface of MICROBLUE showing a few buttons and sliders; right: a user controlling a two-wheel drive robot car using virtual joysticks

MICROBLUE offers a customizable user interface composed of configurable widgets such as joysticks, d-pads, sliders, switches, and text inputs, which map to actuators, sensors, and peripheral electronics; see Fig. 10. This modular approach abstracts the complexities of mobile BLE stack implementation, eliminating the need for dedicated application development and enabling rapid prototyping of control interfaces. In educational robotics contexts, MICROBLUE enhances the experiential learning process by providing an accessible and robust platform for wireless control and parameter tuning, thereby facilitating iterative experimentation and hardware–software integration.

4.4 PROJECTS: Structured Assembly of Modular Robotic Systems. Much like MOTIONGEN and DESIGNER, the PROJECTS[5] is also a web-based application, which provides interactive, stepwise assembly guidance for constructing robotic mechanisms and mechatronic prototypes using SnappyXO hardware kits. Its visualization engine decomposes complex builds into discrete, sequential actions, each supported by annotated 2D/3D part illustrations to ensure precise alignment, orientation, and connectivity during assembly. This structured approach minimizes mechanical errors, supports independent, self-paced learning, and enables repeatable, scalable deployment of design curricula across diverse educational settings.

Functionally, SnappyXO projects serves as a downstream link in an integrated toolchain that includes MOTIONGEN for virtual mechanism synthesis, DESIGNER for custom component creation, and MICROBLUE for BLE-enabled control of assembled robotic systems.

Together, these applications provide an end-to-end workflow that spans ideation, simulation, fabrication, and actuation, allowing for rapid iteration, system-level integration, and experimental testing of robotic designs.

5 Pedagogical Integration and Curriculum Alignment

Integrating a novel hardware–software platform into education requires thoughtful curriculum design and teacher support. From the outset, the development of SnappyXO involved collaboration with educators to develop a standards-aligned, flexible curriculum that leverages its hardware and software capabilities. Each unit in the curriculum engages students in the full design-build-test cycle, thereby reinforcing the iterative nature of engineering. For example, a middle school unit might have students design a mechanism to move an object a certain distance—addressing NGSS MS-ETS1-2 (evaluate competing design solutions). Specific NGSS standards targeted are NGSS 3–5 ETS for elementary school, NGSS 6-8 ETS for middle school, and NGSS 9–12 ETS for high school.

By aligning with such standards, we ensure that teachers can justify the inclusion of robotics projects as a means to teach required concepts, not just as an extracurricular fun activity. In addition, the curriculum targets broader learning outcomes like problem-solving skills, creativity, and teamwork, which are often highlighted in state science standards or the ISTE (International Society for Technology in Education) standards. Each project or challenge in the curriculum comes with clear learning objectives (both content knowledge and skills). SnappyXO is, to our knowledge, the first robotics education product that allows students to truly design, build, and problem-solve their own robotic solutions in alignment with academic standards.

5.1 Curriculum Structure. The SnappyXO curriculum openly available in footnote 3 is organized in a modular fashion, with tracks for different grade bands (upper elementary, middle school, high school, and entry-level college) and different themes (mechanical design, electronics, sensing and actuation, etc.). Early lessons for younger students emphasize fundamental concepts like simple machines (levers, linkages, gears) and basic building skills using the SnappyXO kit. As students progress, the challenges become more complex: e.g., designing a walking mechanism (linkage-based quadruped) in middle school, or a robotic arm with multiple degrees of freedom in high school.

Importantly, the same hardware is used throughout, but the complexity arises from how many parts or what types of mechanisms are involved. This scaffolded approach means a student who learns with SnappyXO in 5th grade can continue to use and build on that knowledge in 9th grade, tackling more advanced projects without having to learn an entirely new kit. Such continuity addresses the common issue of curricula that do not scale between elementary and secondary levels. The curriculum also encourages open-ended projects especially at higher grade levels: rather than all students building the same robot, they might be given a design challenge (e.g., "design a machine that can draw a circle on paper" or "build a device that can lift a cup onto a

[5]http://www.snappyxo.io/projects

shelf") and use an engineering design process to devise and test a solution with SnappyXO. This pedagogical approach mirrors that of design competitions or real engineering scenarios, thereby building students' confidence in tackling unstructured problems.

5.2 Use of Software Tools. A distinctive aspect of SnappyXO's pedagogical integration is the incorporation of the its software applications into the learning sequence. Students often begin a project in the virtual environment of MOTIONGEN: for example, sketching a linkage and simulating its motion to achieve a desired output (like a certain foot trajectory for a walker) or draw the motion path and have the M2M AI engine generate several candidate mechanism design concepts. The software can then suggest a configuration of SnappyXO hardware that best matches the designed mechanism. Students can also refine their mechanism in the software, and even design custom parts using DESIGNER that can be 3D-printed or laser-cut and used with the SnappyXO kit. By combining virtual design with physical construction, students experience a complete design workflow that is very much analogous to what an engineer would do (digital prototyping followed by fabrication). This tight integration of software and hardware is a key pedagogical feature: it teaches computational thinking alongside hands-on building. In a classroom setting, it also opens up more avenues for differentiation; for instance, some students might excel in the digital design aspect, while others focus on building and testing, and then they can come together to share insights. We provide detailed teacher guides on how to use MOTIONGEN in class, including a "guided mode" where the software provides an introduction to students through designing a mechanism that can then be built with the kit. The software's role in the curriculum ensures that coding and math skills are exercised (e.g., adjusting parameters, reading plots of kinematic properties), which complements the hardware's emphasis on mechanical understanding.

5.3 Teacher Training and Resources. Understanding that teachers are key to successful implementation, we developed extensive support materials. This includes lesson plans, editable slide presentations, student worksheets, and solution guides for each project or unit. All instructions are written in a self-paced, student-friendly manner so that teachers can adopt a facilitator role. Professional development workshops were conducted with pilot teachers to gather feedback and refine the materials. One pilot program with elementary educators revealed that, after going through the training and teaching a summer camp with SnappyXO, six out of seven teachers felt confident to continue using and teaching with SnappyXO on their own. This was a crucial outcome: building teacher confidence and capacity.

The availability of premade curriculum also mitigates the typical barrier of "what do I do with this kit?" that often prevents integration of new technology into classes. Moreover, the curriculum was designed to allow flexible implementation. Teachers can use it as a standalone course (e.g., a semester-long robotics elective in middle or high school) or as supplemental modules inserted into science/technology classes. The units are modular, so for example, a teacher could just pick a couple of SnappyXO projects to enrich an existing physics unit on simple machines.

Recognizing the constraints of class time, we also created short video tutorials, project instructions, and examples for students. This proved particularly useful during the COVID-19 pandemic, when some or all learning had to move online. In fact, SnappyXO's curriculum and platform were adapted for fully remote delivery in 2020: students were shipped kits to their homes and instructors taught via video conference, using digital resources to guide builds. Despite the challenges of virtual instruction, teachers reported that students remained motivated and actively collaborated through virtual means. This experience underscored the value of having a well-structured curriculum and support system; even in remote settings, it was possible to facilitate meaningful hands-on learning with SnappyXO.

5.4 Multidisciplinary Learning. An intentional outcome of SnappyXO's curriculum is that it blends multiple STEM disciplines in a natural way. A single SnappyXO project can involve mechanical engineering (designing linkages or structures), electrical engineering (wiring motors and sensors on the robot), and computer science (programming the robot's behavior). For example, a project to build an autonomous rover will require students to design the chassis and drive train (mechanical), connect and power the sensors/actuators (electrical), and write code for the microcontroller to navigate (computer science). By encountering all these elements in one cohesive project, students appreciate the interplay of disciplines—a hallmark of real-world robotics. This integrated approach also caters to different interests: a student drawn to coding might take the lead on that part while another who enjoys building focuses on the mechanism, but both have to communicate and ensure their sub-systems work together. Teachers observed that this led to high engagement: students "bounced ideas off each other" and learned to seek teammates' input to solve problems. The SnappyXO curriculum emphasizes reflection on both successes and failures, prompting students to think about why a design worked or not, thereby reinforcing theory through practice. In pilot implementations, such as a summer program for 32 elementary students, teachers noted that even in an online format, students showed excitement in solving open-ended challenges and took pride in their unique creations. In summary, the pedagogical integration of SnappyXO revolves around providing an authentic engineering experience in an academic framework. The combination of a robust curriculum, teacher support, and alignment with standards makes it feasible for schools to adopt SnappyXO as a regular part of their STEM offerings, not just an extracurricular novelty. The next section will discuss evidence from case studies and pilot programs that validated the educational effectiveness of this approach.

6 Case Studies and Validation

SnappyXO has been deployed and evaluated in a variety of educational settings as part of NSF STTR Phase I and Phase II research [51]. These pilot studies provide insight into the platform's usability, learning outcomes, and impact on student engagement. Below, we summarize key findings from several case studies across K-12 and early college contexts.

6.1 Elementary School Pilot (Riverhead Central School District). In Summer 2020, a pilot program was conducted with the Riverhead Central School District in Long Island involving 32 elementary students (grade 4) and 8 teachers. This program took place during the COVID-19 pandemic and was run fully remotely. Each student received a SnappyXO kit at home, and teachers led the design activities via bi-weekly Google Meet sessions. The curriculum focused on basic mechanical design and robotics concepts suitable for upper elementary level, following a flipped classroom model where instructional resources (videos and guides) were provided on a website for students to review at their own pace. Students were highly motivated by the hands-on nature of the activities, and they demonstrated creativity in their solutions to design challenges. For example, one task asked students to design a mechanism that could "make a character hop"; different teams came up with different linkage configurations to achieve a hopping motion. Teacher feedback was very positive. In pre- and postpilot surveys, six out of seven participating teachers reported that they felt prepared to deliver a SnappyXO-based curriculum in the future and agreed that SnappyXO provided a valuable hands-on engineering experience for their students. The only hesitation from some teachers was the difficulty of doing this virtually—they noted that remote implementation was an obstacle to interaction, and they were eager to teach SnappyXO in person once schools reopened. Once the school reopened for in-person instructions, the SnappyXO program ran again in the summer next year. The Riverhead pilot also provided iterative feedback that was used to refine the curriculum materials (e.g., improving the clarity of instructions for remote contexts). Thereafter, several other elementary schools in Long Island adopted SnappyXO as a solution for STEM enrichment and learning for their students.

6.2 Informal STEM Camp (Long Island Explorium). The Long Island Explorium, a children's science museum and STEM education center, has been running a series of camps using SnappyXO kits for children ages 6–11. According to the Explorium's Director of Education, the SnappyXO framework "worked brilliantly to guide their students into creating more and more advanced builds." She observed that the kit allowed each child to work at their own pace, which is especially beneficial in a mixed-ability group. By the end of the week, campers had built a variety of imaginative creations—from "hopping bunnies" to dancing robots—far beyond what the instructors initially expected for that age group. "I particularly appreciated how the kits allowed each camper to invent, tinker, explore, and create designs that are more sophisticated than you might expect for their age group," the Director noted. Importantly, the success in an informal learning environment like a camp demonstrates SnappyXO's versatility: it can engage young makers in a free-form setting just as well as it can align to standards in a classroom.

6.3 Middle/High School Summer Program. During Phase II, in Summer 2021, the Manufacturing and Technology Resource Consortium (MTRC) of Long Island organized a three-week Design & Robotics summer program using SnappyXO Advanced Robotics Kit for 30 students ranging from middle to high school (ages 12–17). These students came from 29 different school districts across Long Island, reflecting a diverse cross section of backgrounds. The program was conducted in a hybrid format: kits were sent to each student, and instruction happened online due to ongoing pandemic precautions. This program was particularly valuable in testing SnappyXO's scalability and remote delivery at a larger scale. The result was very encouraging—every student was able to actively participate in building and programming robots, even though they were not physically in the same location. We conducted daily check-ins and used breakout rooms for team collaborations. Projects included building a two-wheel differential drive robot with servos in autonomous and remote-controlled mode and eventually ending with a capstone design project. The distributed nature of participants (each in their own home) showed that with the right resources, hands-on PBL can be accomplished remotely: students were mailing photos of their robots, sharing code snippets on Slack, and even doing informal "show-and-tell" demos of their creations over video. This program also served as a stress-test for our online resources—any unclear instructions quickly became evident when students tried to use them independently. We used this feedback to improve the user interface of the SnappyXO projects application and the documentation. At the conclusion of the camp, we collected surveys which indicated high levels of satisfaction; most students said they learned "a lot" about engineering design and would recommend the program to friends. Many had never done a robotics project and they commented on how SnappyXO was challenging but also more rewarding because they "had to figure things out" rather than just follow steps. This aligns with our goal of instilling real problem-solving skills.

6.4 Freshman Design Course (Stony Brook University). SnappyXO was created by the author primarily to satisfy the learning outcomes of his MEC101: Freshman Design Innovation class at Stony Brook University. Even before it became available widely to other students and schools thanks to the support from the National Science Foundation awards, his students have been using it for almost a decade now. MEC101 was created via an Innovation in Instruction and Technology award [52]. In this rather large enrollment class (typically 175+), students learn mechanical design, electronics, and microcontroller programming in the context of engineering design and robotics. To enable problem- and project-based learning and free-up class time, the class is flipped. In this format, students watch the instructor's Khan-academy style videos at home and come to class for just-in-time learning delivered as focused topical presentations followed by in-class activities.

Each home work assignment in this class contains at least one hands-on design and build exercise, often done in a team of two students. For example, in the *Mechanisms and Motion* module, the homework assignment is to design a wiper mechanism on top of a rover, build it using SnappyXO, and then program a microcontroller to move it in a controlled way. The class ends with a capstone robot design project wherein students are allowed to propose a problem to solve. Even at the college level, typical incoming students haven't been exposed to open-ended design in robotics. The instructor of the course reported that using SnappyXO significantly improved the students" understanding of kinematics—something that is hard to convey through theory alone. The unified hardware also means that teams can attempt very different projects (walking robot such as biped, quadruped, hexapod, and toy robots) all with the same kit, which is logistically convenient and cost-effective for the course. This demonstrates potential for SnappyXO's use in higher education, especially in first-year engineering programs that aim to give students a taste of design and robotics. In fact, MEC101 is also offered in several high schools as a way to provide college credit or dual-enrollment engineering courses, which can help with student retention and recruitment into engineering majors. The curriculum for MEC101 is freely available at online[6] under *High-school and College*. Figure 11 shows a few autonomous robot designs by student teams in the MEC101 class.

Across these case studies, a few common themes emerge: (1) students show increased engagement and creativity when given the freedom to design; (2) teachers appreciate the alignment to standards and the improvements in student problem-solving skills; (3) SnappyXO's scalability (both in terms of difficulty and logistical use in different environments) is validated; and (4) the platform lowers barriers to entry for schools new to robotics. In conclusion, the pilot implementations of SnappyXO have provided proof-of-concept that the platform can deliver meaningful educational outcomes. Students from elementary to college level have successfully built and programmed robots of their own design, an experience that most had never had before. In the next section, we compare SnappyXO against existing solutions and discuss its unique advantages in the educational robotics landscape.

7 Discussion: Advantages Over Existing Kits and Value Propositions

This section synthesizes how SnappyXO Design addresses the limitations identified in Sec. 2 and articulates its value propositions relative to existing educational robotics platforms. Rather than reiterating deficiencies of current kits, we focus on how SnappyXO operationalizes design-centric, scalable, and open robotics learning in practice.

7.1 Design-Centric Learning and Creative Divergence. SnappyXO is fundamentally structured as an open-ended design

[6] https://snappyxo.io/curriculum

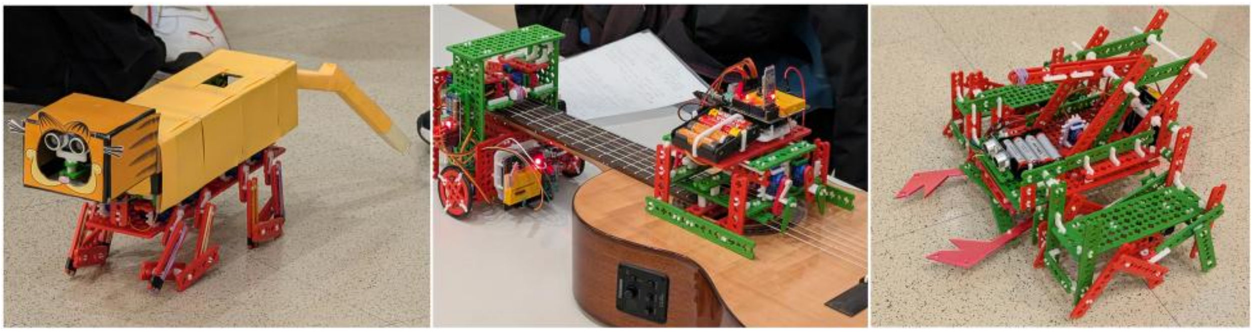

Fig. 11 From left to right: a two-degree-of-freedom legged quadruped, a guitar playing robot, and a one-degree-of-freedom six-bar linkage based walking robot with several other mechanisms

platform rather than an instruction-driven assembly system. Instead of converging students toward predefined robot models, the platform supports problem formulation, mechanism ideation, and iterative refinement with no single "correct" solution. As observed across pilot deployments, groups given the same design challenge routinely produced structurally and kinematically distinct solutions. This divergence is pedagogically significant: it shifts classroom emphasis from execution to reasoning about design trade-offs, constraints, and performance.

A distinguishing feature of SnappyXO is the integration of MOTIONGEN and DESIGNER, which enables students to engage directly with motion synthesis and custom part creation. While many educational kits emphasize programming behavior, few provide tools for designing motion itself. By allowing learners to specify desired motion outcomes and realize corresponding mechanisms physically, SnappyXO supports an authentic engineering workflow that bridges abstract kinematic reasoning and tangible prototyping. This capability enables design experiences typically reserved for advanced undergraduate courses to be introduced earlier and at developmentally appropriate levels.

7.2 Affordability and Programmatic Scalability. Affordability is central to SnappyXO's educational impact. By relying on planar hardware and off-the-shelf electronics, the platform substantially reduces per-student cost relative to mainstream kits. This cost structure enables classroom-scale deployment, one-to-one student access, and reuse of the same hardware inventory across multiple grade levels. In contrast to age-specific platforms, SnappyXO supports a vertically integrated learning pathway in which increasing sophistication arises from project complexity rather than from changing kits.

Scalability also extends to logistics and maintenance. Replacement parts are inexpensive and, in many cases, locally fabricable using common makerspace tools. Standard batteries, open microcontrollers, and design software further reduce total cost of ownership and simplify long-term adoption. These factors collectively lower barriers for schools seeking to integrate robotics beyond enrichment settings and into core instructional time.

7.3 Open Architecture and Technical Authenticity. SnappyXO's open architecture distinguishes it from closed, vendor-specific ecosystems. Mechanically, the standardized hole-and-clip interface accommodates a wide range of commercially available sensors, actuators, and controllers. Pedagogically, this flexibility allows instructors to tailor projects to curricular goals and student readiness while exposing learners to components and workflows commonly encountered in higher education and industry.

On the software side, SnappyXO does not constrain users to a single programming environment. Students may progress from block-based interfaces to text-based programming in C/C++ or PYTHON using the same hardware, supporting continuity in computational skill development. This openness reduces platform obsolescence and aligns with maker-oriented and research-informed STEM pedagogies that value extensibility and experimentation.

7.4 Educational Efficacy and Evidence From Practice. SnappyXO was developed through iterative research and deployment, incorporating project-based learning principles and sustained educator feedback. As demonstrated in the case studies presented in Sec. 6, students across age groups were able to design and build original robotic systems, while teachers reported increased confidence in facilitating open-ended engineering activities. These outcomes suggest that the platform supports learning goals beyond competition performance, emphasizing design reasoning, collaboration, and systems thinking.

By combining low cost, design expressivity, and curricular support, SnappyXO broadens participation in robotics education, particularly in contexts with limited resources. Its value lies not only in technical capabilities, but in enabling learning experiences that are difficult to realize with conventional kits.

Summary. Taken together, SnappyXO offers a cohesive solution that integrates design-centric hardware, generative software tools, and standards-aligned curriculum within an open and affordable framework. Rather than replacing all existing platforms, it complements competition-oriented or advanced fabrication systems by emphasizing early and sustained engagement with engineering design. In this sense, SnappyXO occupies a distinctive position at the intersection of learning tool and invention kit, supporting authentic, scalable, and adaptable robotics education.

8 Conclusions and Future Work

This article presented SnappyXO Design, an NSF-funded translational robotics platform aimed at revolutionizing STEM education through a design-first approach. SnappyXO's hardware kit introduces a unique combination of planar modular parts and compliant snap-fit connectors, enabling students to build a vast array of robots and mechanisms with a minimal inventory of components. We detailed how the kit's technical innovations—from its alphabet-shaped clips that enforce geometric and kinematic constraints to its integration with common electronics—provide an open-ended yet structured way for learners to engage in authentic engineering design.

Alongside the hardware, the SnappyXO ecosystem includes a motion design and simulation software, as well as a comprehensive curriculum for K-12 and beyond, forming an end-to-end solution for educators to implement project-based learning in robotics. Through case studies spanning elementary schools, STEM camps, high school programs, and even college courses, we validated SnappyXO's educational impact. The results indicate that students using SnappyXO gain not only coding skills, but also a deeper understanding of mechanical design and systems thinking, learning to prototype, test, and iterate like real engineers. Moreover, SnappyXO's affordability and scalability mean that it can be deployed in contexts where robotics was previously impractical, thereby democratizing access to quality hands-on STEM learning. In essence, SnappyXO offers a balanced blend of being a research-driven educational innovation and a ready-to-use classroom tool, aligning with the goals of translational robotics to bring lab innovations to real-world users.

In future, we plan to expand MOTIONGEN synthesis coverage to broader mechanism classes (including 3D), deepen integration between synthesized mechanisms and manufacturable part export, and continue advancing UI to integrate advanced functions and features to support industry users. On the hardware side, we will add multiplane motion transmission with extendable shafts and constrained rotational joints for power transfer across planes. Moreover, we will achieve manufacturing consistency via injection molding all the parts (in addition to laser cutting) that will remove laser taper ("asymmetric fit"), increase dimensional consistency, and enable tuning material properties (e.g., "softer material for clips" to adjust flexibility).

Acknowledgment

This publication has research support from various National Science Foundation awards: CMMI (# 1563413) and I-Corps award (#1823736) to Stony Brook University and STTR phase I (# 1843476) and Phase II award (# 2126882) to Mechanismic Inc with a subcontract to Stony Brook University. We also acknowledge funding support from the SUNY Research Foundation, NY-State Strategic Partnership for Industrial Resurgence, SensorCAT, and Fuzehub as well as the Institute for Advanced Computational Sciences at Stony Brook University.

Conflict of Interest

The author co-founded Mechanismic Inc. until it was acquired in February 2026. Mechanismic Inc. received the NSF STTR funding and funded research in his lab through a subcontract. The terms of this arrangement have been reviewed and approved by Stony Brook University in accordance with its policy on objectivity in research.

Data Availability Statement

No data, models, or code were generated or used for this article.

References

[1] Wiens, G. J., 2025, "Robotics: Road Ahead for Mechanical and Physical Necessities, Bridging Artificial Intelligence, and Integration Challenges," Lett. Trans. Rob., **1**(1), p. 014001.

[2] The Science & Technology Action Committee, 2024, *The U.S. Needs More STEM Workers ASAP. A National Science and Technology Strategy Can Help.*, The Science & Technology Action Committee. https://sciencetechaction.org/news-item/theus-needs-more-stem-workers-asap-a-national-science-and-technology-strategy-can-help/

[3] Beer, R. D., Chiel, H. J., and Drushel, R. F., 1999, "Using Robotics to Teach Science and Engineering," Commun. ACM, **42**(6), pp. 85–92.

[4] Eguchi, A., 2014, "Robotics as a Learning Tool for Educational Transformation ," Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, Padova, Italy, July 18.

[5] Khanlari, A., 2013, "Effects of Educational Robots on Learning Stem and on Students' Attitude Toward STEM," 2013 IEEE 5th Conference on Engineering Education (ICEED), Kuala Lumpur, Malaysia, Dec. 4–5, pp. 62–66.

[6] Manyika, James, Lund, Susan, Chui, Michael, Bughin, Jacques, Woetzel, Lola, Batra, Parul, Ko, Ryan, and Sanghvi, Saurabh, 2017, *Jobs lost, jobs gained: What the future of work will mean for jobs, skills, and wages*, McKinsey Global Institute. https://www.mckinsey.com/featured-insights/future-of-work/jobs-lost-jobs-gained-what-the-future-of-work-will-mean-forjobs-skills-and-wages

[7] Angel-Fernandez, J., and Vincze, M., 2018, "Towards a Formal Definition of Educational Robotics," Proceedings of the Austrian Robotics Workshop 2018, Innsbruck, Austria.

[8] Ching, Y.-H., and Hsu, Y.-C., 2023, "Educational Robotics for Developing Computational Thinking in Young Learners: A Systematic Review," TechTrends, **68**, pp. 423–434.

[9] Trapero-González, I., Hinojo-Lucena, F. J., Romero-Rodríguez, J.-M., and Martínez-Menéndez, A., 2024, "Didactic Impact of Educational Robotics on the Development of Stem Competence in Primary Education: a Systematic Review and Meta-Analysis," Front. Edu., **9**, p. 1480908.

[10] Phokoye, S. P., Epizitone, A., Nkomo, N., Mthalane, P. P., Moyane, S. P., Khumalo, M. M., Luthuli, M., and Zondi, N. P., 2024, "Exploring the Adoption of Robotics in Teaching and Learning in Higher Education Institutions," Informatics , **11**(4).

[11] Anwar, S., Bascou, N. A., Menekse, M., and Kardgar, A., 2019, "A Systematic Review of Studies on Educational Robotics," J. Pre-College Eng. Edu. Res. (J-PEER), **9**(2), p. 2.

[12] "LEGO Group Lego Spike Prime," https://spike.legoeducation.com/prime/lobby/, Accessed July 26, 2025.

[13] VEX Robotics, 2025, "Vex Robotics Kits," https://www.vexrobotics.com/, Accessed July 26, 2025.

[14] Pitsco Inc, 2025, "Tetrix Robotics Set," https://www.pitsco.com/collections/tetrix-robotics, Accessed July 26, 2025.

[15] Fischertechnik GmbH, 2017, "Fischertechnik," http://www.fischertechnik.de/en/Home.aspx, Accessed July 26, 2025.

[16] Makeblock Co. Ltd, 2025, "Makeblock," www.makeblock.com, Accessed July 26, 2025.

[17] Purwar, A., 2018, "NSF I-Corps: A Design Driven Educational Robotics Framework." Award No. 1823736.

[18] Purwar, A., Ge, Q. J., and Godse, A., 2020, "Modular Frame Connectors and System Utilizing Same," U.S. Patent #10,561,958, SUNY Research Foundation, http://www.snappyxo.com.

[19] Ge, Q. J., Zhao, P., and Purwar, A., 2013, "Decomposition of Planar Burmester Problems Using Kinematic Mapping," *Advances in Mechanisms, Robotics and Design Education and Research. Mechanisms and Machine Science*, Vol. 14, V. Kumar, J. Schmiedeler, S. Sreenivasan, and H. Su, eds., Springer, Heidelberg.

[20] Ge, Q. J., Zhao, P., Purwar, A., and Li, X., 2012, "A Novel Approach to Algebraic Fitting of a Pencil of Quadrics for Planar 4R Motion Synthesis," ASME J. Comput. Inform. Sci. Eng., **12**(4), p. 041003.

[21] Zhao, P., Li, X., Purwar, A., Thakkar, K., and Ge, Q. J., 2013, "A Task Driven Approach to the Synthesis of Spherical 4r Linkages Using Algebraic Fitting," Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 6A: 37th Mechanisms and Robotics Conference, Portland, OR, Aug. 4–7.

[22] Li, X., Zhao, P., Purwar, A., and Ge, Q. J., 2013, "A Task Driven Approach to Simultaneous Type Synthesis and Dimensional Optimization of Planar Parallel Manipulator Using Algebraic Fitting of a Family of Quadrics," Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 6B: 37th Mechanisms and Robotics Conference, Portland, OR, Aug. 4–7.

[23] Zhao, P., Purwar, A., and Ge, Q. J., 20134, "On the Complete Synthesis of Finite Positions With Constraint Decomposition via Kinematic Mapping," Proceedings of the ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 5A: 38th Mechanisms and Robotics Conference, Buffalo, NY, Aug. 17–20.

[24] Ge, Q., Purwar, A., Zhao, P., and Deshpande, S., 2017, "A Task-Driven Approach to Unified Synthesis of Planar Four-Bar Linkages Using Algebraic Fitting of a Pencil of G-Manifolds," ASME J. Comput. Inform. Sci. Eng., **17**(3), p. 031011.

[25] Purwar, A., Deshpande, S., and Ge, Q. J., 2017, "Motiongen: Interactive Design and Editing of Planar Four-Bar Motions via a Unified Framework for Generating Pose- and Geometric-Constraints," ASME J. Mech. Rob., **9**(2), p. 024504.

[26] Ge, Q. J., Ge, X., Purwar, A., and Li, X., 2015, "A Null-Space Analysis Method for Solving Bilinear Equations in Kinematic Synthesis of Planar and Spherical Dyads," Proceedings of the 14th IFToMM World Congress, Taipei, Taiwan, Oct. 25–30, pp. 322–328.

[27] Sharma, S., Purwar, A., and Ge, Q. J., 2019, "An Optimal Parametrization Scheme for Path Generation Using Fourier Descriptors for Four-Bar Mechanism Synthesis," ASME J. Comput. Inform. Sci. Eng., **19**(1), p. 014501.

[28] Sharma, S., Purwar, A., and Ge, Q. J., 2019, "A Motion Synthesis Approach to Solving Alt-Burmester Problem by Exploiting Fourier Descriptor Relationship Between Path and Orientation Data," ASME J. Mech. Rob., **11**(1), p. 011016.

[29] Li, X., Zhao, P., Purwar, A., and Ge, Q., 2018, "A Unified Approach to Exact and Approximate Motion Synthesis of Spherical Four-Bar Linkages Via Kinematic Mapping," ASME J. Mech. Rob., **10**(1), p. 011003.

[30] Ge, X., Purwar, A., and Ge, Q. J., 2016, "From 5-SS Platform Linkage to Four-Revolute Jointed Planar, Spherical and Bennett Mechanisms," Vol. Volume 5B: 40th Mechanisms and Robotics Conference of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, p. V05BT07A081.

[31] Ge, X., Purwar, A., and Ge, Q. J., 2017, "Finite Position Synthesis of 5-SS Platform Linkages Including Partially Specified Joint Locations," Vol. Volume 5B: 41st Mechanisms and Robotics Conference of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, p. V05BT08A074.

[32] Zhao, P., Li, X., Purwar, A., and Ge, Q. J., 2016, "A Task-Driven Unified Synthesis of Planar Four-Bar and Six-Bar Linkages with R- and P-Joints for Five-Position Realization," ASME J. Mech. Rob., **8**(6), p. 061003.
[33] Li, X., Ge, X., Purwar, A., and Ge, Q. J., 2015, "A Unified Algorithm for Analysis and Simulation of Planar Four-Bar Motions Defined with R- and P-Joints," ASME J. Mech. Rob., **7**(1), p. 011014.
[34] Deshpande, S., and Purwar, A., 2022, "A Unified Approach to Dyad and Triad Synthesis for Planar Mechanisms for Motion Generation," Proceedings of the 2022 USCToMM Symposium on Mechanical Systems and Robotics, Rapid City, SD.
[35] Sharma, S., and Purwar, A., 2020, "Using a Point-Line-Plane Representation for Unified Simulation of Planar and Spherical Mechanisms," ASME J. Comput. Inform. Sci. Eng., **20**(6), p. 061002.
[36] Deshpande, S., and Purwar, A., 2019, "A Machine Learning Approach to Kinematic Synthesis of Defect-Free Planar Four-Bar Linkages," ASME J. Comput. Inform. Sci. Eng., **19**(2), p. 021004.
[37] Deshpande, S., and Purwar, A., 2019, "Computational Creativity via Assisted Variational Synthesis of Mechanisms Using Deep Generative Models," ASME J. Mech. Des., **141**(12), p. 121402.
[38] Deshpande, S., and Purwar, A., 2020, "An Image-Based Approach to Variational Path Synthesis of Linkages," ASME J. Comput. Inform. Sci. Eng., **21**(2), p. 021005.
[39] Nurizada, A., Dhaipule, R., Lyu, Z., and Purwar, A., 2025, "A Dataset of 3m Single-Dof Planar 4-, 6-, and 8-Bar Linkage Mechanisms With Open and Closed Coupler Curves for Machine Learning-Driven Path Synthesis," ASME J. Mech. Des., **147**(4), p. 041702.
[40] Lyu, Z., and Purwar, A., 2024, "Point-Based Models: A Unified Approach for Geometric Constraint Analysis of Planar N-Bar Mechanisms with Rotary, Sliding, and Rolling Joints," ASME. J. Mech. Rob., **17**(4), p. 044514.
[41] Deng, X., Nurizada, A., and Purwar, A., 2025, "Path Synthesis of Spatial Revolute-Spherical-Cylindrical-Revolute Mechanisms Using Deep Learning," Front. Mech. Eng., **20**(2), p. 9.
[42] Purwar, A., and Chakraborty, N., 2023, "Deep Learning-Driven Design of Robot Mechanisms," ASME J. Comput. Inf. Sci. Eng., **23**, pp. 1–13.
[43] Lyu, Z., Purwar, A., and Liao, W., 2024, "A Unified Real-Time Motion Generation Algorithm for Approximate Position Analysis of Planar N-Bar Mechanisms," ASME J. Mech. Des., **146**(6), p. 063302.
[44] Nurizada, A., and Purwar, A., 2024, "Transforming Hand-Drawn Sketches of Linkage Mechanisms Into Their Digital Representation," ASME J. Comput. Inform. Sci. Eng., **24**(1), p. 011010.
[45] Nurizada, A., and Purwar, A., 2023, "An Invariant Representation of Coupler Curves Using a Variational AutoEncoder: Application to Path Synthesis of Four-Bar Mechanisms," ASME J. Comput. Inform. Sci. Eng., **24**(1), p. 011008.
[46] Sharma, S., and Purwar, A., 2022, "A Machine Learning Approach to Solve the Alt-Burmester Problem for Synthesis of Defect-Free Spatial Mechanisms.," ASME J. Comput. Inf. Sci. Eng., **22**(2), p. 021003.
[47] Halwah, A., and Purwar, A., 2025, "Generative Design of Planar Four-Bar Motions," ASME J. Mech. Rob., **18**(1), p. 011005.
[48] Purwar, A., and Ge, Q. J., 2025, "Advancements in Kinematic Synthesis: Data-Driven Methods for Mechanism Design," ASME J. Comput. Inf. Sci. Eng, **25**(12), p. 120810.
[49] Kaipa, K., Purwar, A., Ayala, O., and Bawab, S., 2025, "Transforming Engineering Education: Project-Based Learning and Technology Integration in a Senior-Level Mechanisms Course, ASEE 2025 Annual Conference & Exposition, Paper # 49376," 2025 ASEE Annual Conference & Exposition, Montreal, Quebec, Canada, June 22–25.
[50] Purwar, A., 2023, "Machine-Learning Driven Robot-Motion Design: Introducing a Web-Based Mechanism Design Software."
[51] Purwar, A., Liao, W., Kale, A., and Chuang, P., 2019–2024, "Robotics Launchpad - A Design-Driven Educational Robotics Framework," $1,295,000 Award No. 1843476, 2126882 (National Science Foundation STTR Phase 1 and 2).
[52] Purwar, A., Ge, Q. J., and Aceves, P., 2014, "Freshman Design Innovation: Suny Innovative Instruction Technology Grant (IITG)," $60,000.