Control and Communication in Modular Robots: Distributed and Centralized Schema for Localization and Reconfiguration - A Review

Vyasraj R. Vignesh *
Henry M. Gunn High School, Palo Alto, CA 94306
vyasrajvignesh@gmail.com
July 31, 2026

Abstract

Modular robot systems are composed of modules - either as homogeneous modules equipped with similar sensors, actuators, docking linkages, communication, power, and compute or increasingly as a heterogeneous system with specialized modules. Modules can locomote and reconfigure into different morphologies in time and space, to work under diverse structures, tasks, terrains and meta-goals such as safety, and self-repair. Unlike monolithic robots structured for a specific task, modular robot systems are treated as high degree-of-freedom (dof) distributed systems with a distinct control and communication problem space. This paper reviews the tradeoffs within control and communication schemes for such systems, in their support for two specific capabilities – (1) robot *localization*, where unlike a monolithic robot, the system must localize many n-dof modules, and shares the problem space with mobile and multi-robot systems, and (2) robot *reconfiguration* – a problem space unique to modular robot systems, where a module subset must be identified, motion-planned, and closed-loop controlled until a target configuration is achieved. We seek to determine how varied levels of module heterogeneity, autonomy and communication channel properties like latency and noise, lead to different choices for control layer colocation. Related system requirements such as low communication latency for real-time responsiveness and safety, and higher compute and power for iterative n-dof configuration-space solvers – too present competing tradeoffs in a system with latency-bound central controllers, and limited on-module resources.

Keywords Modular robotics · Robot control architectures · Robot localization · Robot reconfiguration

1 Introduction

Modular robots [1] that are reconfigurable under centralized or distributed control or even autonomously, have emerged over the last decade as a viable class of engineered robotic systems. For mission objectives and task environments where morphology needs to be variable and the system (structure, kinematics, power, compute, and sensorimotor functions) must adapt to varying conditions, modular robot systems provide versatility, reusability and resilience [2]. Compared to fixed-function, fixed-morphology robots, modular robots are able to modify and additively or subtractively augment their morphology and connectivity, for control or reconfiguration. Such reconfiguration is executed proactively or reactively to changing task requirements, emerging environmental constraints such as terrain, damage or failure, and measured impairments in system resources such as power, compute, and communication bandwidth. While early research was shape-driven [3], emphasizing the building-block morphology and linkage properties of modular robots, recent research has emphasized the task-driven design of such systems [4].

In a modular robot system (Figure 1), the modules can be isomorphic and isofunctional, each carrying independent power, sensors and end-effectors that connect together in different topologies to perform tasks [5] or they can be heterogeneous, with autonomous compute, locomotion, and communication with specialized roles [6] and functions. In recent decades, sophisticated schemes for reconfiguration [7], self-reconfiguration [8], self-assembly [9], and self-repair [10] have been studied [11] and demonstrated under centralized and distributed command and control [12]. These mechanisms with intelligently-learned rulesets [13], policies and transition-models [14] have equipped modular robot

* Currently, a rising high-school senior, studying linear algebra and differential equations, with developing interests in system science and control engineering across mechanical, aerospace and electrical systems.

systems with environment-responsive proprioceptive [15] and exteroceptive faculties [16] that enable diverse-tasking and taskgroup-functions, shared linkage, power, compute, and improved degrees of resilience [17], redundancy, and fault-tolerance. Autonomous goal-driven reconfiguration leading to self-repair and self-assembly remains an active frontier of research, especially for aquatic, aerospace, and off-planet missions.

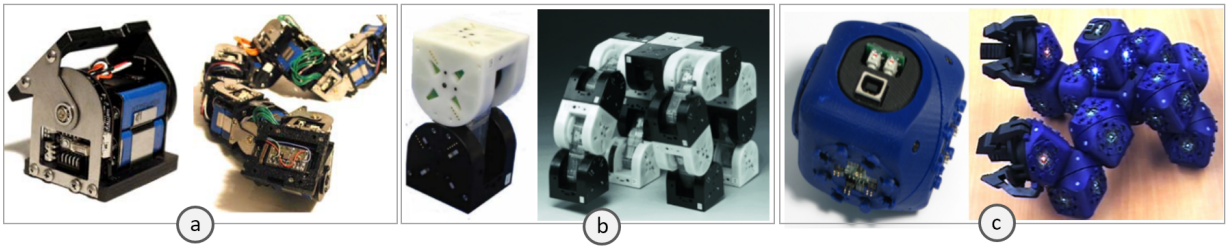

Figure 1: Modular robots – module + system (a) PolyBot [18] (b) M-TRAN [19] (c) Molecubes [20]

As modular robots [21] are tasked for real-world missions, capable control, compute, and communication subsystems [22] are required to achieve the desired system subdomain outputs. These subdomains include navigation (localization, motion-planning), locomotion, assembly (identification, recruitment), linkage (reconfiguration, docking), and manipulation (forward and inverse kinematics, end-effector control). Within these subdomains, multiple schemata for communication and control have been studied and implemented in the research literature. In this paper, we review these schemes and their effectiveness, specifically for localization and reconfiguration (Figure 2). Before a modular robot system can begin a reconfiguration, the system must determine its *localization* – the position and orientation of its modules within a workspace, floor or geographical region. Localization itself feeds into a larger question – that of configuration-discovery where the shape and connectivity-graph of the modules is sensed and computed as adjacency-matrices in real-time [23]. A *reconfiguration* plan must then be computed, which could require a new cycle of motion-planning, locomotion, and linkage – at different hierarchies or scales of the system – intra-module, inter-module and global. The plan must then be *communicated* to or between segments or clusters of the modular robot system. As reconfiguration proceeds, closed-loop control cycles in an iterative, sense-plan-act cadence must execute at hierarchical system scales to ensure stability and convergence towards the desired end configuration. A large-dof (degree-of-freedom) system like a modular robot, presents specific design, scale, synchronization, latency and optimization constraints, forcing researchers to adopt distributed local optima solutions that are then linked via boundary-conditions into a global, suboptimal but feasible solution [24].

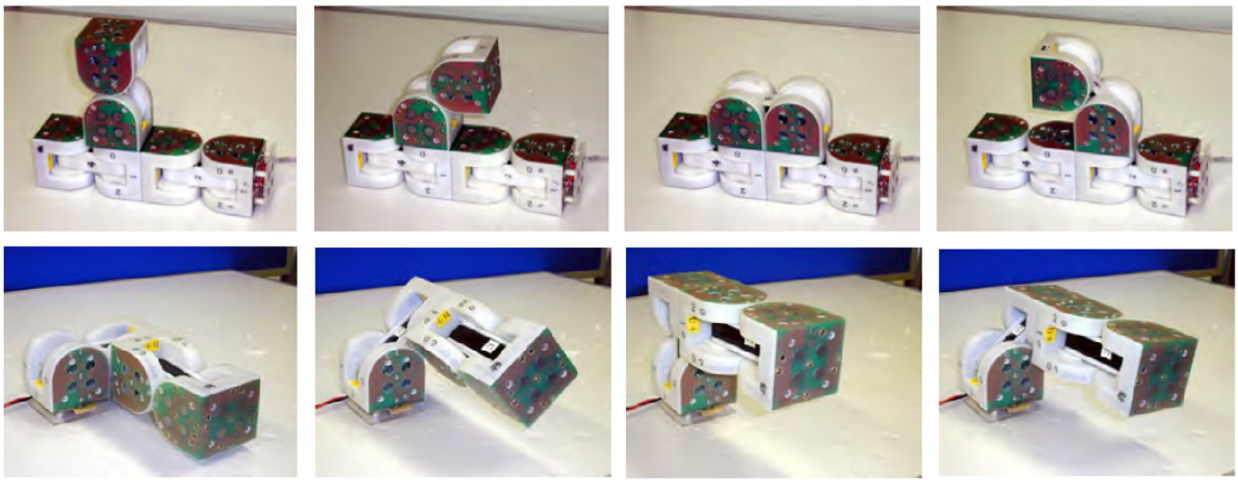

Figure 2: Modular robot reconfiguration [19] – forward-roll motion (top), morphology change (bottom)

As an exemplar of the multifaceted nature of the control and communication design space, communication itself demands several contextual considerations. In a modular robot system, heterogeneous modules can be role-assigned - such as workers (modules with actuators and manipulators), monitors (with sensors), controllers (hosting storage and compute for closed-loop control), or connectors (providing structural elements for linkage and articulation). To serve the needs of such role-assigned modules, communication can be local (among neighbor-linked modules), or

clustered (among geo-fenced modules) or global (across all modules). Global communication, in broadcast or multicast form, effected within a hop-to-hop, hierarchical or mesh-network, via active wireless signaling, or passive lightpath or infrared signaling – might appear to be a necessary and sufficient solution, but its advantages and constraints (latency, power) don't make it a universal fit, especially when a proportion of control itself needs to be on-module (local) to meet low-latency, and real-time bounds. Communication [25], as a medium for command and control, must serve the needs of physical control algorithms for motion, actuation, and reconfiguration, as well as higher-level logical control algorithms for leader-election, for platooning, recruitment for cluster assembly, sensor-fusion, and on-device energy management. Varied sensorial data originated at the module - (scalars such as location, joint angles, and power reserves) or (vectors such as optical/electromagnetic camera images) present different requirements on communication bandwidth, flow-control, error-rates, interference, and noise tolerance. Paired with varied control data addressed to the module (task-plan, microcontroller programs, transition-model or state-action table updates), the communication scheme to support a predetermined frequency and volume of information required by a coupled control scheme (distributed, global, open-loop or closed-loop), must operate under specific constraints determined by distance, distribution, on-board storage, and on-board power. Wireless communication has emerged as the preferred medium and open-source, middleware-based approaches [26] with pluggable communication and control stacks, both in hardware/firmware, and software have gained ground in the last decade.

Communication serves control, and in this regard, control in modular robot systems presents an upper-level design and constraint space. The control system governs how modules communicate, where control decision or computation occurs, how failures are reported post-sensing, and how reconfiguration, repair, and assembly is commanded and controlled thereafter. With *centralized* control, a master module or external controller issues commands, receives, and holds state and coordinates other modules, in a hub-spoke topology. In a *decentralized* system, control is entrusted to multiple master modules, who may have singular or shared functions among them. With *distributed* control, each module executes its own control loop and coordinates with peers. Among distributed systems, recent architectural proposals include fully distributed (no hierarchy [27] and federated (cluster masters with local module autonomy) [28] schemes. These schemes apply to a broader, multi-agent, decision-making problem space with meta-concerns like learning, policy, optimization, and security in the mix. For modular robot systems though, a much smaller schema-subset applies – where they are modeled as distributed IoT systems with scalability and resilience needs.

The choice of control system impacts capabilities upward in planning and downward into hardware. From low-level joint-torque regulation for module-reconfiguration – to be executed in milliseconds – to high-level trajectory execution for module-assembly – spanning seconds or minutes – control systems are built from separate logical and hierarchical components with different sense-plan-act (SPA) cycle time constraints. Far evolved from the classical PID (proportional-integral-derivative) control system, several paradigms for non-linear control have emerged over the years. Early approaches [29] were based on adaptive model-based control [30], which uses an explicit dynamic model that undergoes continual calibration from online sensor data and requires sufficient communication and on-board storage resources – these were suitable for optimal control of small-dof monolithic robots with few manipulators or locomotion actuators like wheels or legs. These are inadequate though for large-dof, distributed or autonomous robot systems like modular robots. The next step was model-predictive-control (MPC) [31] that continually solves an optimization problem, requires high compute resources on the controller, and is suited more for fixed-function, monolithic robots. Variants of MPC that tune certain parts of MPC at the expense of other parts have emerged and have been shown to be more suited for modular/articulated robot systems. These include joint-space-control [32], which is computationally simpler but limits task specification flexibility for the module and task-space control [33], that requires real-time Jacobian inversion computation on the module or on the controller. By their very construct, modular robots are limited-multi-agent systems where *reactive* control alone is inadequate. Instead, *deliberative* schemes [34] have been proposed ranging from behavior arbitration [35] to state-augmented [36] systems. The kernels for such control processes are thematically similar - the kernel reads in sensor data and writes out actuator commands; in between, an arbitration layer selects or blends sensor data and modulates out-commands following a computed or predefined policy. These out-commands can target multiple behaviors in a single control cycle. For instance, a module must be controlled for power-management, obstacle-avoidance, terrain-following, and bounds-detection for its manipulator's dexterous/reachable workspace, all under a high-level operating state such as task-execution, locomotion, reconfiguration or assembly.

Control schemes that require computation that is iterative until convergence or that need to be executed off-module in a centralized controller, are infeasible or unsafe in modular robot missions where real-time obstacle response, unpredictable environments and high communication latency are extant. In such contexts, behavior-based control schemes [37] are employed where a central planning or arbitration module is not waited upon for a state-action decision [38]. When control loop latency cannot exceed a mission limit such as 100 milliseconds or when the operational environment is stochastic and cannot be modeled apriori, centralized control becomes infeasible. Modules experiencing faults must fail, degrade or disassemble gracefully, desirably under local control, to retain fault-isolation and safety. On the other hand, (a) when missions require multi-step, sequential or graph-based planning across multiple time/cycle

horizons (b) where emergent behavior previously unmodeled is likely to surface and (c) where task dependencies require state-sharing and state-updates that exceed module-local decision boundaries, arbitrated, centralized control becomes a necessity. The tradeoff here is a breach of the real-time constraint envelope that is mandated by safety and fail-close rules. This has forced robot system designers to include low-latency approaches at the logical [39] and physical layer [40], and then combine local, reactive controls, with centralized deliberative controls to capture the advantages of both paradigms.

Structure of this paper

We begin by reviewing how communication and control systems have evolved and are currently architected for distributed cyber-physical systems. While several system classes fall under this designation - mobile warehousing robots, fixed IoT systems to swarm robots - we focus on modular robot systems as our class of study. We seek to understand how varied levels of module heterogeneity, autonomy, and communication channel properties like latency and noise, lead to different choices for control layer colocation – on-module, distributed near-*edge*, and centralized. **Section 2** discusses general control and communication considerations for modular robot systems. **Section 3** focuses on the specific properties and requirements of the *localization* function in modular robots. The output of localization feeds into a *motion-planning* layer where a planned trajectory must be executed (with close-loop control) for designated modules to locomote and assemble to a system state ready for reconfiguration. **Section 4** studies the communication and control requirements for *motion-planning*: its real-time, two-way, low-latency, requirements on the communication channel and its distributed control with local, cluster-level, and centralized components. **Section 5** focuses on the *reconfiguration* action in modular robot systems. After motion-planned assembly has completed, reconfiguration begins with docking and ends with linkage. This is executed among closely-spaced modules where near-field wireless signaling (NFC) becomes possible, local infrared and electro-inductive techniques like RFID can work, and simple paired-inter-module control loops running locally on-module, become necessary and feasible. The paper concludes with a summary of the tradeoffs inherent in the design of control-communication for modular robot systems and currently active and emerging research in this area.

2 Communication for distributed control

As modular robot systems have grown in heterogeneity and complexity, with diverse taskability, and reconfigurability, the capabilities required of the control and communication system (Figure 3) have dramatically increased. In the recent past, (Mendoza, Schultz, Stoy [41] reviewed the tradeoffs inherent in local versus global communication and control where the choice was parameterized on robot topology. The paper concluded that while generally global communication was necessary for centralized control, and local communication was required for distributed control, robot topology influenced the choice as well. Low-connectivity, chain-type topologies are distinguished from high-connectivity, lattice-type configurations with global communications preferred for the former type. However, the context of this conclusion was for communication done hop-by-hop across linked mechanical interfaces. In prior research, wired, infrared [42], and sonar [43] mediums for communications have been studied and applied. But with limitations arising from line-of-sight and mechanical alignment of transmitters/receivers, these have fallen out of favor over the decade. These systems were far removed from modern wireless networks that favorably provide global, pan-module addressing and reach, but have different tradeoffs of their own. Wireless mesh networks [44] degrade in capacity for large-diameter, module-connectivity-graphs but surprisingly retain constant capacity for cubic-lattice module configurations.

Modular robot systems are expected to be able to change their morphology self-sufficiently and progressively for control, reconfiguration, and repair purposes. Bespoke control architectures for systems with specific morphologies have been studied [45] but research continues to seek a generalized architecture. Consider two premises: (a) general closed-loop control algorithms designed for an n-dof system are complex and not suitable for modules (with limited resources and capabilities) operating in environments that are similarly resource-constrained (communication latency, noise, compute, power) and (b) modular robots are necessarily evolutionary, that is - they are expected to change morphology (such as chains, lattices, trusses), constitution (number of modules and per-module-dof) and linkage. At a basic level, this makes the number of inputs and outputs for a control system variable, making it difficult to optimize or even bound the complexity and resource-use (compute, power, communication) of such a control system. Without such bounds, it becomes infeasible to analytically apportion the centralized and decentralized parts of such a control system. Studies recognizing this issue [46] have found that general decentralized approaches independent of size (for a given morphology), were operatively feasible over a larger variety of sizes, as expected, but struggled under changing morphology within the same system. The underlying cause is that the system is designed and tuned apriori with heuristics learned through offline simulation and becomes an overfit for the morphology it was optimized or trained for. A threshold change in morphology uncovers convergence and stability problems that undermine latency and safety guarantees.

Exploring the design and optimization space for control, researchers have studied adjacent, near-range, and local-neighborhood control as well as global system-wide control for navigation, locomotion, and assembly. To support such varied control schemes, and with powerful on-board microcontrollers, communications based on standard protocols and software implementations within a ROS (robot operating system)-based robotics middleware [26] have emerged. A system architecture that serves modular and mobile robot systems is described in (Wolff, Schulz) [47]. An early paper [48] introduces the middleware approach, and a later paper [49] develops its design principles. The modern control model is that of a distributed system with real-time constraints, derived from task-driven requirements on performance, safety, and fault-tolerance. Real-time constraints require some portion of the controller to be implemented on distributed compute hardware for low latency, safety, and fault-tolerance. Policy-based control requires a different portion of the controller to be implemented as software on a fixed-function microcontroller, or field-programmable (FPGA) hardware. Safety policies (failover, failclose, failopen) must have dedicated, time-constrained reaction pathways for the (sense, policy-rule-select, and actuation) feedback loop.

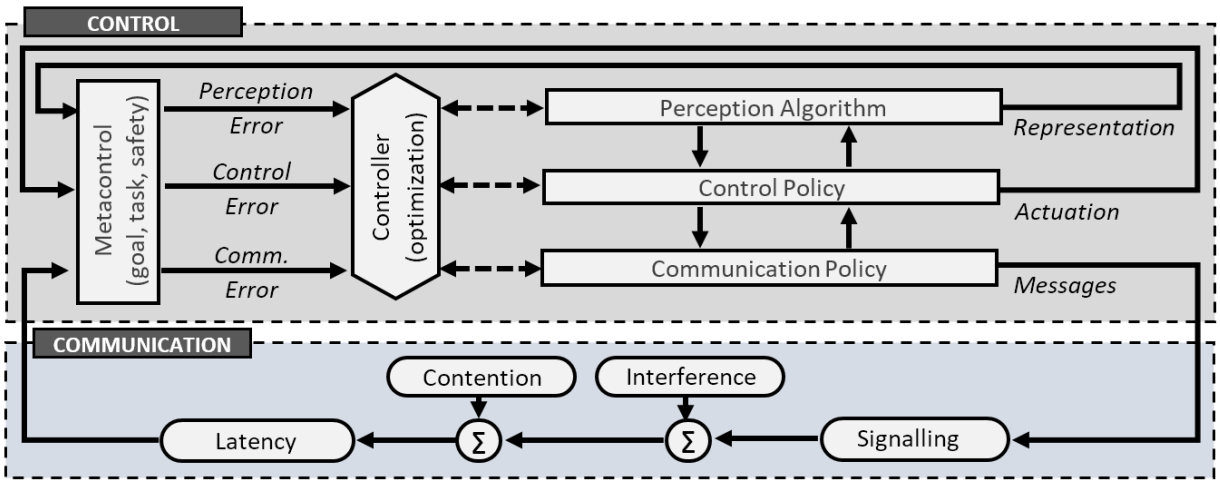

Figure 3: Control and communication in a multi-robot system - adapted from (Gielis) [25]

In the last decade, the emergence of IoT platforms [50] and next-generation cellular wireless networks like 5G [51] have further expanded the communication capacity and function available to mobile [52] and modular robot systems. Specific subprofiles, like 5G IoT (Internet of Things), now support low-latency with adequate bandwidth for applications where continuous sensing and actuation is required. Within the 5G IoT subsystem, service profiles like URLLC (ultra-reliable low-latency communication) [40] have been developed that support hard, real-time responsiveness, and reliability. These were originally developed for applications in autonomous driving, robotic surgeries, and electrical power-grid control. Unlike short-range wireless networks like ZigBee [53] and Wi-Fi (IEEE 802.11), 5G provides for a much larger geographical range. This expanded range, with high speeds and low-latency, enables compute-intensive control algorithms to be run on power-intensive controllers that can be consolidated and located, further away from the field of operation [54], further mitigating constraints arising from having less-capable controllers on the module itself with limited power and compute. A wider geographic range introduces issues of addressing, channel ownership and interference in a dense network of IoT systems. Unlike Wi-Fi, where a network identifier (SSID) is created at a base station and explicitly joined to by nodes, 5G IoT provides for multiple ad hoc, dynamic, and transient subnetworks, hosted within the same base station coverage area and dynamically segmented and joined to by nodes. With network slicing of this form [55], a cluster of modules identified to undergo a reconfiguration can subscribe to and join a single segmented private subnetwork. Simpler broadcast-based algorithms can be used to talk to all modules in the cluster, without requiring specific addressing or multicast within the cluster. Furthermore, this segmentation avoids imposing a broadcast signal on modules outside the cluster that would have to expend on-board power to receive, inspect, and reject such a signal. For modular robot systems, 5G IoT does have drawbacks. Because 5G uses high-frequencies and multiple channels to achieve its low-latency and reliability properties, its cost and power consumption profile remains unviably high for modular robot systems with a large number of modules with limited per-module power and cost budgets. Hybrid approaches that combine lower-cost 3G with a 5G overlay [56] have been studied.

A fundamental tension in distributed system control is control bandwidth [57], measured against computation cost. A module undergoing reconfiguration, assembly or end-effector actuation on a task, imposes a required datarate bandwidth for its sense-plan-act feedback control loop. For instance, a module requiring 100 Hz in control supervision, limits the time available per cycle, and consequentially mandates and/or constrains the compute resource needed to satisfy the limit.

Specialized [58] or additional compute units increase cost and power demands. The high control rate itself precludes many candidate control algorithms. Control schemes with simple table lookups (policy rulesets, transition-models) can sustain high control rate, for multiple, concurrent targets (actuators), while a decision-tree or graph-based scheme that must run sequentially or a model-predictive-control scheme [59] that must run an n-dimensional iterative solver until convergence, are inherently infeasible. The communication fabric that ties the controllers and modules in modern middleware-based robotics, of which modular robots are one species, is invariably based on commercial packet-radio networks like ZigBee, WPAN (IEEE 802.15.3) and BLE (Bluetooth Low Energy). For such networks, the time window available per control cycle must account for communication latency, time for error-correction, packet-loss and retransmission. Control logic placement is categorized as either on-module or near-edge where ruggedized compute is positioned within a few milliseconds of network distance, that further bounds the sense-plan-act cycle time. An example control pipeline implemented in middleware like ROS2 [12], begins with sensor ingestion (from force-torque sensors, position, and angle encoders, IR/optical cameras), followed by local onboard or near-edge preprocessing (sampling, quantization, noise filtering), followed by centralized calculation of state-space model coordinates (localization) that are eventually fed into a motion-planning component that generates the final control commands. While local, onboard controllers provide autonomy, they are constrained by (power and weight), while near-edge control on dedicated hardware is free of those constraints but adds network latency. Unlike fixed-function monolithic robots, large-dof modular robots with assembly and reconfiguration abilities, require complex, closed-loop, compute-intensive control schemes that prefer near-edge controllers. However, safety requires that some autonomous control reside onboard to provide compliance, force-limiting behaviors, and deterministic response times independent of the communication network or controller distance.

3 Communication and control for robot localization

Locomotion, assembly, and reconfiguration in mobile modules of a modular robot system depends on a continually, reliable mechanism to determine current location and orientation (pose) of the constituent modules – a system function termed *localization* (Figure 4) [60]. Computer-vision based localization, introduced in (Atiya, Hager) [61] and later applied using artificial landmarks has been shown to be viable [62], but requires more onboard compute power; is unstable under ambient lighting changes, is sensitive to camera-calibration parameters and is not accurate enough for navigation. Passive open-loop control can be achieved with coded infrared light maps [63]. However, for flexible closed-loop motion control under a global controller, with high accuracy, stability, robustness, and real-time sense-compute-control cycles, wireless sensor networks, (at least as basic as RFID) [64] with mesh architectures are necessary. The fine, spatial resolution needed for a modular robot system operating in a limited-volume environment, requires each module's receiver to triangulate signals from at least three ephemeris reference transceivers based on SSI (signal-strength-indicator), ToA (Time-of-arrival), AoA (Angle-of-arrival), and TDoA (Time-difference-of-arrival). Short distances and high electromagnetic wave speeds require extremely time-sensitive calculations that are easily affected by issues like multipath fading, interference and non-LOS (line-of-sight) conditions that affected early wireless networks. These challenges have been largely addressed with modern, low-power, limited-area wireless network standards.

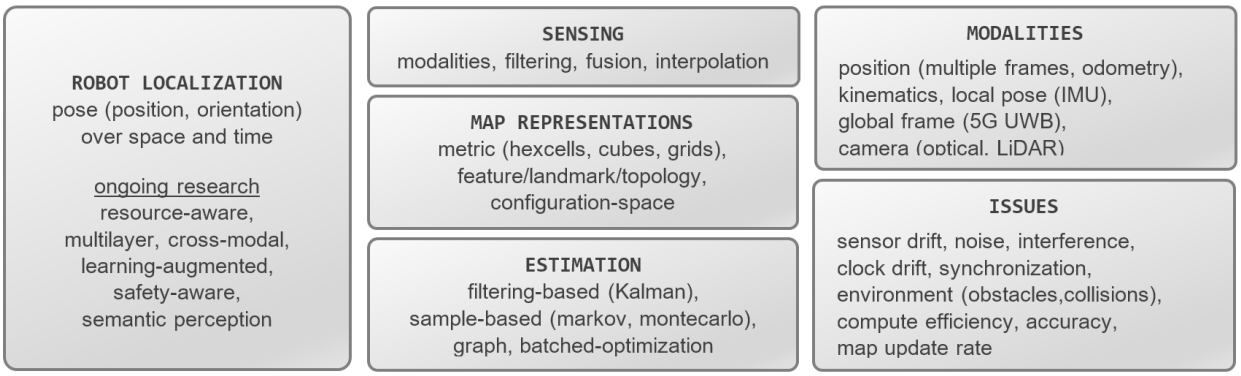

Figure 4: Robot Localization [60] – problem and solution spaces

Among modern, short-range, wireless networks, the ZigBee [53] standard was originally developed for industrial and home automation. Early experiments demonstrated localization using the ZigBee network with artificial infrared landmarks [65]. ZigBee is designed for short-range monitoring and control and is suitable for global open-loop and closed-loop control for navigation and positioning; however modules in a modular robot system also require

locally-paired, near-range communication for reconfiguration, docking, and assembly tasks. Recent research [66] has demonstrated localization with millimeter-scale accuracy using NFC (Near-Field Communication) that was not achieved with earlier RF-based protocols. A hybrid of RF-based long-range signaling for global decentralized control and infrared(IR)-based, near-range, self-synchronized pairing among adjacent modules has been studied [67], but the overall negatives of IR rule out its reliable use in modular robots. IR is highly susceptible to blockage, misalignment, phase-drift, and interference with indoor-LED lighting. IR has been successfully employed for paired systems [68] where a few global IR cameras localize a few, in-field-of-view units such as virtual-reality (VR) headsets. However, despite early headways [69], IR has not gained acceptance and use for a spatially-distributed system where a global IR camera must reliably localize multiple modules from their IR beacons. Similar concerns apply to motion-capture (MoCap) systems originally built for biomechanical studies and CGI (computer-generated-imagery). These systems have been studied for robot localization [70], but being designed for humanoid form-factors, they require external, optical field-array cameras and onboard IMUs (inertial measurement units) that are impractical for modular robots.

Above the hardware layer, distributed multi-robot localization [71] has been studied as a separate system function wherein both local and global components have been identified as essential for field robots. A localization solution that relies less on centralized and global, closed-loop control and compute resources is much desirable – at least to meet minimal guarantees for real-time responsiveness for safety and collision-avoidance. A distributed approach [72] that first lets neighbor modules compute relative-localization with respect to a referent module, results in a set of overlapping local transform frames. These frames are then fused or patch-worked into a global coordinate space by the centralized controller. This approach enables neighbor modules to execute their trajectories independently over incremental path segments until the next position or orientation waypoint, without requiring central, low-latency, high-bandwidth closed-loop control. These approaches have been termed as *cooperative* localization and have been studied under limited connectivity [73], asynchronous [74], sparse [75], and unreliable [76] communication contexts. Decentralized fusion methods that stitch relative and absolute position measurements into batched updates [77], also aim to reduce communication costs and reliance on connectivity, but theoretical accuracy and convergence estimates are yet to be field-tested. These subtechniques fall under a larger class of SLAM (simultaneous localization and mapping) methods, and are being actively researched [78].

Robot localization, specifically for modular robots, invariably includes both a centralized component and distributed, cluster-local or module-local components. In a monolithic robot, the singular robot body need only know its localization with respect to a global frame (map) and can receive, compute, and execute a motion or task plan within that context. For modular robots with distributed localization, each module may know its own localization within multiple frames – the global frame, the cluster-local frame such as a frame attached to a hub node of a lattice structure or the module-local frame. For simpler modules, a centralized controller senses from, computes and communicates localization to all modules. Modern localization, however, is increasingly embedded within a larger SLAM system [79], where sensor data (affected by drift, noise) is fused from multiple sources, undergoes filtering (particle, Kalman) and feature-extraction, followed by pose-graph compute using iterative solvers, and finally resulting in a topological map representation with occupancy grids. For all modules in each occupancy grid, the central controller then redistributes their localization information. Each module, now under its own *odometer* frame, will use dead-reckoning and its IMU to trace the next segment of its trajectory, until sensor drift triggers a new frame refresh. The communication, control and compute needed to support these schemes is an order of magnitude larger than modular robot systems of the previous decade, and necessitate programmability and a building-block architecture at the firmware and software layers of the controller. Localization with LiDAR systems [80], using infrared wavelengths, where the interference and noise of IR is compensated with multiple sensors, additional filtering-compute, and stochastic estimation, is resource-intensive and suited for high-unit-cost mobile or multi-robot systems but not yet for modular robots where modules are intended to have a lower unit-cost profile.

Despite the sophistication and optimality achieved by bespoke methods for specific localization problem classes, a more practical approach using current commercial middleware such as ROS [81], configured to interface with preselected sensor, communication, compute, and control components, remains the current viable and general approach for localization in heterogeneous modular robot systems. A system intended for structured, static environments – where low-latency in localization and localization-guided locomotion is needed – where simpler localization must be equipped locally on-module – is different from a system intended for an unstructured, dynamic environment requiring multiple sensors, generic compute for stochastic, nonlinear tracking techniques and frequent, short-cycle localization remapping. These middleware components impose additional power, bandwidth, and compute requirements but costs are factored as system costs and a price to pay for their redundancy, programmability, and flexibility benefits.

4 Communication and control for robot motion-planning

Motion planning and execution as a core robotics system function [82], enables modular robots achieve the benefits and versatility of being modular and provides additional capacities for assembly, docking, reconfiguration, fault-isolation, and repair. Motion planning is complex for modular robots with each n-dof module making up a large-dof system. The underlying mechanics includes goal, obstacle and constraint representation, path search, trajectory parameterization, and execution feedback that applies not just to one robot, but several modules that must coordinate to achieve a target docking, assembly or configuration. This inherently requires two-way communication and off-module, near-edge compute, and control. Since motion planning produces velocity, force, and distance outputs, this control component is responsible for enclosing its actions within safety, recovery, and repair envelopes.

Motion-planning (Figure 5) [83] for modular robot systems has been studied under varied logical control schemes. Early control schemes studied decentralized motion-planning as an overlay [84] or a complement [85] to an established centralized scheme – the goal being to reduce communication and compute cost and to avoid a single point of failure. The context however was structured environments, pre-defined tasks or homogeneous modules [86], that afford hierarchical planning at lower compute cost. In modern research, goals have shifted towards unstructured environments [87] and heterogeneous modules (multi-robots) with dynamically-defined tasks [88]. A centralized, global planning [89] component continues to remain necessary, where algorithms to optimally explore the configuration space must compute trajectories and select module motion gaits. For a modular robot system, motion planning occurs under two separate domains – *locomotion* and *reconfiguration*. Before a motion plan to effect a reconfiguration can be created, current localization, morphology and configuration must be sensed, modeled and recognized. Based on this recognition, a motion-plan executes first in the locomotion domain, where target modules must locomote to and assemble into a cluster, ready for reconfiguration. Next, a motion plan in the reconfiguration domain executes, where each module is plan-controlled to dock and articulate into the desired configuration.

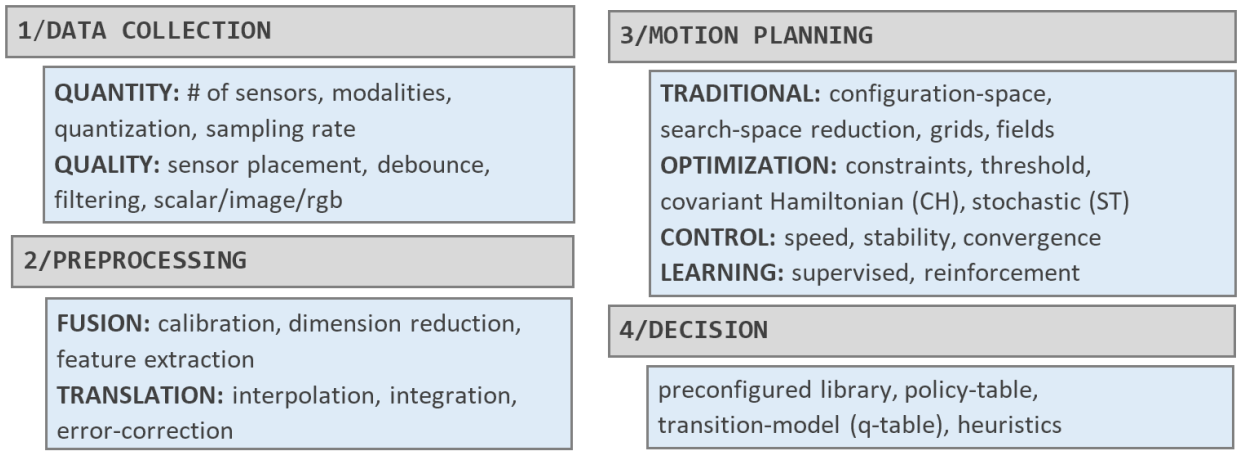

Figure 5: Motion planning steps – adapted from (Zhou, Huang, Fränti) [83]

For modular robot systems, motion-planned reconfiguration from an initial configuration to a target configuration desired for the next task or goal step, involves searching the configuration space for a feasible, optimal sequence, parametrized by a set of metrics [90]. This effort, framed as a combinatorial optimization problem or a Markov decision process [88], becomes exponentially harder and intractable with higher per-module-dof and number and heterogeneity of modules. Constraints imposed by workspace (reachable) perimeter, and time (responsiveness, safety) further limit the degree of optimality (or even feasibility). If a reconfiguration cannot be planned and executed within a bounded space and time envelope, the system must treat the goal as a no-solution and abstain, instead of executing and incurring safety and collision risks that would require a systemwide configuration reset. Varied approaches to address this problem – precomputed library [4], hierarchical [86], layered [91], incremental [92] – which trade optimality or versatility for speed – have been studied, but an overarching theme persists. As modular robot system designers aim to build greater versatility and resilience for varied missions by endowing very many modules with novel dof, this ends up inherently increasing the compute, control, power and time needed for reconfiguration-planning and execution, consequently leading to limits in system size and heterogeneity.

Distributing the planning solution into global and local controllers with modern middleware and on-module processing and control with piecewise execution (one module or cluster at a time) has been shown to be practical in commercial

systems. Centralized motion planners that must return a result within a bounded cycle window must buffer their trajectory output, feeding the real-time control loop one batch at a time. Current research points to a hybrid approach where distributed motion planners are segmented – different local planners are assigned to specific modular functions, like those that plan for structural modules, or locomotion modules, and can be situated much closer to the affected modules making NFC (near-field) communication possible. A centralized planner produces a global motion plan, with trajectory, waypoints, velocity profiles, and deviation thresholds. This plan, once pushed to the individual modules by a communication network, must be held and executed over incremental, short time-horizons with local closed-loop control by a local planner that incorporates obstacle avoidance, compliance and safety overrides. In a modular robot system, the individual modules move in relation to each other; so the world model (includes obstacles) is not static post-plan. Requiring each module to receive and memoize every other module's current and future trajectory is expensive and is not scalable in terms of communication and compute. Computing and communicating more state requires higher bandwidth, lower latency and higher storage, compute, and power. More state causes longer control latency, which in turn affects safety and failclose/failopen response times.

To bound time, latency and resource-use, a segmented or clustered approach[93] to motion planning is adopted where a module is recruited (tagged) into a specific cluster based on shortest paths between the modules and the cluster. The per-module localization information needed to determine the cluster neighborhood, can either be stored on the central planning controller and updated frequently by each module, or can be determined dynamically by a broadcast query that the centralized controller issues to all modules which then reply back individually. Since the former approach requires sufficient communication bandwidth and processing, at both module and controller for frequent updates, the latter approach is preferred. A study[21] of the effect of cluster size on reconfiguration motion control reveals additional tradeoffs. Here, the size of each cluster is in terms of either the number of modules per cluster or the spatial volumetric bounding-box occupied by the cluster. As these clustered or teamed modules are controlled to follow a compatible motion plan to achieve the target assembly or configuration, incremental planning on the time-scale at the central controller, and segmented plans per-cluster for tagged modules produces piecewise optimal solutions – but lower path quality and suboptimal trajectories at the system level. However, the payoff is adaptable paths over time and space, and is an acceptable price to pay to contain computational and communication loads for an n-module modular robot system.

5 Communication and control for robot reconfiguration

Modular robot systems are assemblies or configurations of modules where each module is function-assigned with preset sensorimotor interfaces, with the ability to be reconfigured [94], detached or replaced independently. At a higher-level, the ability to reconfigure or self-reconfigure [2] is studied as evolutionary modular robotics [95] within a given class of morphologies [96], where a reconfiguration sequence involves multiple modules assembling together and forming linkages [97] to fulfil a structural, locomotion, or task objective. An assembly sequence precedes the reconfiguration phase, where individual modules are enumerated, and a neighborhood of modules is recruited and programmed to reconfigure. A pairwise docking sequence is then executed under local control to complete the linkages. A module, by definition, embodies a discrete unit with sense, actuation, compute, control, and power functions, but must submit to a centralized or distributed control function executed off-unit to coordinate its operation with other modules for a common goal. In heterogeneous modular robot systems, the modules may be incapable of locally executing tightly-coupled, reactive control loops because the modules themselves may have disparate abilities (such as a sensor module, a structural linkage module, and an actuator module) that must configure together in the required order to effect a task goal. Each module of the triad must be independently but coordinatedly controlled. The algorithms and mechanisms for reconfiguration also vary by geometry, morphology and the motion primitives available at each module. Studies of reconfiguration in two-dimensional planar spaces [98], or with specific module shapes such as cuboids [99] or with specific connectivity topologies such as lattices [100], have expectedly yielded specific solutions, but are not generally extendable to support reconfiguration across varied modular robot system classes.

Before reconfiguration can occur, modular robot systems must first establish their current configuration. This is built incrementally from continual localization data received and reconciled over time at a central controller, and results in a configuration recognition step. Alternatively, a configuration discovery phase is initiated only when the need arises. Configuration is explicitly concerned with topology – the compatibility and connectivity of module pairs. Compatibility asks if unpaired modules are functionally (suitable for task) and structurally (proximity, docking interfaces) pairable, while connectivity asks what pairings currently exist that need to be unpaired and commanded for a new pairing. Distributed control is imperative in this context, and inherently introduces tradeoffs that require a hybrid approach [23] to mitigate. Using short-range IR (highly directional, line-of-sight dependent) for port-port intermodule communication and long-range ZigBee (omnidirectional) for global communication is a compromise with notable concerns – IR is susceptible to crosstalk that needs localized frequency modulation; IR can drain power from continuous RF pulsing and needs active control to bound sleep and wake states to limit power use; ZigBee's mesh network overhead grows

quadratically with the number of modules, and can exceed real-time control cycle limits; explicit design is needed to ensure that only changes in state (and not the full state) are propagated over ZigBee and only after the state has been sufficiently established locally - after debouncing and after stability has been verified post active movement.

Reconfiguration itself presents equally challenging tradeoffs for communication and control, requiring bespoke approaches to balance system resources across the centralized vs decentralized divide [101]. The connectivity graph of modules changes, and so does a module's neighborhood. A module's membership of a cluster or a linked-group changes as well. Coordinating communication for identification and recruitment must also occur before a reconfiguration can start. This implies that modules not in close NFC-range or IR-range must be able to communicate. Wireless communication automatically becomes the only viable choice for this pre-reconfiguration phase. For close-proximity wireless networks, Bluetooth [102] and ZigBee are established industry standards with commercially available components – hardware in multiple form-factors at various price/performance ratios, and open-source software and firmware libraries that integrate into higher-level middleware blocks. However, each wireless system unfortunately presents different tradeoffs. Bluetooth, originally built for consumer devices, requires a statically configured or dynamically-discovered central controller with limits on network size (number of nodes) and consumes more power, but provides higher bandwidth and longer range. ZigBee, on the other hand, is designed for low-power and shorter range, suffices for low-bandwidth data exchange, and allows a large network size. It achieves this feat [103] by requiring nodes to form an ad hoc wireless mesh network, with each node acting as a repeater or a router to extend range with no central controller and has self-healing [104] abilities (timed fail-over and fail-back for broken links) due its mesh nature.

Reconfiguration merely produces the initial state for a new configuration. Subsequent challenges include maintaining connectivity as the task is performed, and resetting if a motion constraint, or collision with an obstacle causes loss of topology. Modular robot modules have limited onboard power; their linking actuators have limited torque and can yield due to overwhelming environmental forces or yield for intentional compliance designed to limit damage, prevent failure or preserve safety. A control diffusion scheme [105] – where a central controller selects a post-reconfiguration reactive-behavior table for each module – where each module is controlled locally by this natively-stored table – enables each module to autonomously assume an acceptable failstate in case topology is lost. In modular robot systems with many modules, even from a few tens to a hundred, a reconfiguration plan executed one or a few modules at a time is impractical due to long completion times and low resource utilization. Early studies [106] introduced parallel reconfiguration where for a given configuration change, a suitable partition [107] of the module set is first computed. These disjoint subsets of modules are first reconfigured in parallel [108] to a set-local final state, with the subsets then connected serially to a global final state. This requires separate module addressing, and channel allocation in the communication layer, and a control layer with MIMO-based model predictive control in FPGA hardware [58], or multithreaded software control in microprocessor-based middleware [109]. This sophistication is necessary to enable dynamic sizing of communication, compute and control resources since the number of partitions and the cardinality of each partition can vary with the system size (number of modules), topology (lattice, chain, tree), configuration delta (difference between start and end configuration) and the environment (space, motion and time constraints).

6 Conclusion

Modular robot systems continue to advance in research and implementation, along several fronts – heterogeneity, morphologies, sensors, communication media, and control architectures. As distributed, large-dof cyberphysical systems, they present multifaceted, nonlinear problem spaces in localization, motion-planning, and configuration, for which no universally optimal and efficient communication and control architecture is expected to exist. Different morphologies, on-module resources, task profiles, and reconfiguration capabilities engender complex control schemes that vary in how they balance the tradeoffs inherent in centralized vs. distributed colocation of control while remaining feasible within available power and latency envelopes. Abstract research control schemes can obscure field issues in communication (variable wireless network quality arising from noise and contention) and control (iterative, convergence-dependent control algorithms that grow superlinearly with the number of modules, in compute and power needs) thus worsening response-latency and weakening temporal guarantees required for safe, compliant operation. These structural tradeoffs are best addressed with standard, middleware-based system design, distributed control, with piecewise, incremental and concurrent schemes that recognize the fact that a modular robot system is a multiunit system, with limited power, compute, and latency budgets for every control cycle.

Approaches designed for mobile or multi-robot systems cannot be effectively transplanted on to modular robots. A control architecture for a mobile robot cannot be scaled to a legion of modules, or that for a heterogeneous multi-robot cannot be applied to homogenous modules - all while meeting responsive control cycle times and power usage. The current outlook increasingly favors and points towards a programmable middleware stratum – communication with ad hoc, wireless mesh networks, and distributed control with a centralized controller able to run compute-and-time intensive algorithms, with programmable on-module logic for local closed-loop control. Future research is expected to identify a

universal architectural framework – horizontal layers, vertical slices within a layer, that define digitally-twinned unit component blocks that can act as proving grounds for exploratory designs. This system substrate can then support flexible task and configuration schemes above it and novel physical module designs below it, thus producing practical systems that move beyond simulation towards real-world modular robot platforms.

References

[1] M. Yim, Y. Zhang, and D. Duff, "Robotics: Modular robots," *IEEE Spectrum*, vol. 39, no. 2, pp. 30–34, Feb. 2002, ISSN: 0018-9235. DOI: 10.1109/6.981854

[2] M. Yim et al., "Modular self-reconfigurable robot systems," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 43–52, Mar. 2007, ISSN: 1070-9932. DOI: 10.1109/mra.2007.339623

[3] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-tran: Self-reconfigurable modular robotic system," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 431–441, 2002. DOI: 10.1109/TMECH.2002.806220

[4] G. Jing, T. Tosun, M. Yim, and H. Kress-Gazit, "Accomplishing high-level tasks with modular robots," *Auton. Robots*, vol. 42, no. 7, pp. 1337–1354, Oct. 2018, ISSN: 0929-5593. DOI: 10.1007/s10514-018-9738-1

[5] G. Liang, D. Wu, Y. Tu, and T. L. Lam, "Decoding modular reconfigurable robots: A survey on mechanisms and design," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 740–767, 2025. DOI: 10.1177/02783649241283847

[6] J. Baca, M. Ferre, and R. Aracil, "A heterogeneous modular robotic design for fast response to a diversity of tasks," *Robot. Auton. Syst.*, vol. 60, no. 4, pp. 522–531, Apr. 2012, ISSN: 0921-8890. DOI: 10.1016/j.robot.2011.11.013

[7] A. Brunete, A. Ranganath, S. Segovia, J. P. de Frutos, M. Hernando, and E. Gambao, "Current trends in reconfigurable modular robots design," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 1 729 881 417 710 457, 2017. DOI: 10.1177/1729881417710457

[8] K. Stoy, D. Brandt, and D. J. Christensen, *Self-Reconfigurable Robots: An Introduction* (Intelligent Robotics and Autonomous Agents). Cambridge, MA: MIT Press, 2010, ISBN: 978-0262013710.

[9] E. Bray and R. Groß, "Recent developments in self-assembling multi-robot systems," *Current Robotics Reports*, vol. 4, pp. 1–16, Nov. 2023. DOI: 10.1007/s43154-023-00106-y

[10] R. Alattas, S. Patel, and T. Sobh, "Evolutionary modular robotics: Survey and analysis," *Journal of Intelligent & Robotic Systems*, vol. 95, Sep. 2019. DOI: 10.1007/s10846-018-0902-9

[11] A. Elkady and T. Sobh, "Robotics middleware: A comprehensive literature survey and attribute-based bibliography," *Journal of Robotics*, vol. 2012, May 2012. DOI: 10.1155/2012/959013

[12] E. Munera, J.-L. Poza-Lujan, J.-L. Posadas-Yague, J. Simo, and J. F. B. Noguera, "Distributed real-time control architecture for ros-based modular robots," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 233–11 238, 2017, 20th IFAC World Congress, ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2017.08.1600

[13] J. Whitman, M. Travers, and H. Choset, "Learning modular robot control policies," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 4095–4113, 2023. DOI: 10.1109/TRO.2023.3284362

[14] M. Wirkus, S. Arnold, and E. Berghöfer, "Online reconfiguration of distributed robot control systems for modular robot behavior implementation," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1283–1308, 2020. DOI: 10.1007/s10846-020-01234-9

[15] B. Hosseinkhani Kargar, K. Miras, and A. E. Eiben, "Exploring proprioceptive feedback in the evolution of modular robots," in *Parallel Problem Solving from Nature – PPSN XVIII*, Cham: Springer Nature Switzerland, 2024, pp. 405–418, ISBN: 978-3-031-70071-2. DOI: 10.1007/978-3-031-70071-2_25

[16] J. Whitman and H. Choset, "Learning modular robot visual-motor locomotion policies," *arXiv preprint arXiv:2210.17486*, 2022. DOI: 10.48550/arXiv.2210.17486

[17] T. Cordie, J. Roberts, M. Dunbabin, R. Dungavell, and T. Bandyopadhyay, "Enabling robustness to failure with modular field robots," *Frontiers in Robotics and AI*, vol. 11, p. 1 225 297, 2024, ISSN: 2296-9144. DOI: 10.3389/frobt.2024.1225297

[18] A. Golovinsky, M. H. Yim, Y. Zhang, C. Eldershaw, and D. Duff, "Polybot and polykinetic system: A modular robotic platform for education," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2004. DOI: 10.1109/ROBOT.2004.1308017

[19] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Self-reconfigurable modular robot m-tran: Distributed control and communication," in *The Proceedings of First International Conference on Robot Communication and Coordination*, ser. RoboComm '07, Athens, Greece: IEEE Press, 2007, ISBN: 9789639799080. DOI: 10.4108/icst.robocomm2007.2119

[20] V. Zykov, P. Williams, N. Lassabe, and H. Lipson, "Molecubes extended: Diversifying capabilities of open-source modular robotics," in *IROS-2008 Self-Reconfigurable Robotics Workshop*, 2008.

[21] K. Gilpin and D. Rus, "Modular robot systems," *IEEE Robotics & Automation Magazine*, vol. 17, pp. 38–55, Oct. 2010. DOI: 10.1109/MRA.2010.937859

[22] B. Salemi and M. Moll, "Superbot: A deployable, multi-functional, and modular self-reconfigurable robotic system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 3636–3641. DOI: 10.1109/IROS.2006.281719

[23] J. Baca, B. Woosley, P. Dasgupta, and C. A. Nelson, "Configuration discovery of modular self-reconfigurable robots: Real-time, distributed, ir+xbee communication method," *Robotics and Autonomous Systems*, vol. 91, pp. 284–298, 2017, ISSN: 0921-8890. DOI: 10.1016/j.robot.2017.01.012

[24] M. Moussa, B. Piranda, A. Makhoul, and J. Bourgeois, "Cluster-based distributed self-reconfiguration algorithm for modular robots," in *Advanced Information Networking and Applications*, Cham: Springer International Publishing, 2021, pp. 321–332, ISBN: 978-3-030-75100-5. DOI: 10.1007/978-3-030-75100-5_29

[25] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi-robot systems," *Current Robotics Reports*, vol. 3, no. 4, pp. 213–225, 2022. DOI: 10.1007/s43154-022-00090-9

[26] D. Wildgrube Bertol, G. Fink, M. Marchitto, Y. Nisticò, M. Pestarino, and C. Semini, "Dls2: A distributed microcomponent-based robotic software architecture," *Results in Engineering*, vol. 32, p. 112 048, 2026, ISSN: 2590-1230. DOI: 10.1016/j.rineng.2026.112048

[27] A. Miele, M. Lippi, and A. Gasparri, "A distributed framework for integrated task allocation and safe coordination in networked multi-robot systems," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 11 219–11 238, Jan. 2025. DOI: 10.1109/TASE.2025.3532023

[28] J. Weber, M. Gurtner, A. Lobe, A. Trachte, and A. Kugi, "Combining federated learning and control: A survey," *IET Control Theory & Applications*, vol. 18, pp. 2503–2523, 18 2024. DOI: 10.1049/cth2.12761

[29] H. Kawasaki, T. Bito, and K. Kanzaki, "An efficient algorithm for the model-based adaptive control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 496–501, 1996. DOI: 10.1109/70.499832

[30] O. Castillo and P. Melin, "Intelligent adaptive model-based control of robotic dynamic systems with a hybrid fuzzy-neural approach," *Applied Soft Computing*, vol. 3, no. 4, pp. 363–378, 2003, Soft Computing for Control of Non-Linear Dynamical Systems, ISSN: 1568-4946. DOI: 10.1016/j.asoc.2003.05.007

[31] S. Katayama, M. Murooka, and Y. Tazaki, "Model predictive control of legged and humanoid robots: Models and algorithms," *Advanced Robotics*, vol. 37, no. 5, pp. 298–315, 2023. DOI: 10.1080/01691864.2023.2168134

[32] M. Bhardwaj et al., "Fast joint space model-predictive control for reactive manipulation," *CoRR*, vol. abs/2104.13542, 2021. DOI: 10.48550/arXiv.2104.13542

[33] N. Torres Alberto, A. Skuric, L. Joseph, V. Padois, and D. Daney, "Model Predictive Control for robots adapting their task space motion online," working paper or preprint, Apr. 2023. [Online]. Available: https://inria.hal.science/hal-04073876

[34] R. Chatila, "Deliberation and reactivity in autonomous mobile robots," *Robotics and Autonomous Systems*, vol. 16, no. 2, pp. 197–211, 1995, Moving the Frontiers between Robotics and Biology, ISSN: 0921-8890. DOI: 10.1016/0921-8890(96)81009-8

[35] L. Iocchi, D. Nardi, and M. Salerno, "Reactivity and deliberation: A survey on multi-robot systems," in Springer Nature, Jul. 2001, vol. 2103, pp. 9–32, ISBN: 978-3-540-42327-0. DOI: 10.1007/3-540-44568-4_2

[36] F. Ingrand and M. Ghallab, "Deliberation for autonomous robots: A survey," *Artificial Intelligence*, vol. 247, pp. 10–44, 2017, Special Issue on AI and Robotics, ISSN: 0004-3702. DOI: 10.1016/j.artint.2014.11.003

[37] F. Michaud, M. Nicolescu, and M. J. Matari, "Behavior-based systems," in *Springer Handbook of Robotics*, Springer, 2016, pp. 307–328. DOI: 10.1007/978-3-319-32552-1_13

[38] B. G. Woolley, G. L. Peterson, and J. T. Kresge, "Real-time behavior-based robot control," *Autonomous Robots*, vol. 30, no. 3, pp. 233–242, 2011. DOI: 10.1007/s10514-010-9215-y

[39] J. Luck, P. L. McDermott, L. Allender, and D. C. Russell, "An investigation of real world control of robotic assets under communication latency," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (HRI '06)*, Salt Lake City, Utah, USA: Association for Computing Machinery, 2006, pp. 254–261, ISBN: 1-59593-294-1. DOI: 10.1145/1121241.1121277

[40] F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz, D. Pensky, and A. Knoll, "5g for robotics: Ultra-low latency control of distributed robotic systems," in *2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, 2017, pp. 69–72. DOI: 10.1109/ISCSIC.2017.27

[41] R. F. Mendoza-Garcia, U. P. Schultz, and K. Støy, "On the efficiency of local and global communication in modular robots," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2009, pp. 1502–1508. DOI: 10.1109/IROS.2009.5354011

[42] A. Castano, A. Behar, and P. Will, "The conro modules for reconfigurable robots," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 403–409, 2002. DOI: 10.1109/TMECH.2002.806233

[43] R. Grabowski, L. E. Navarro-Serment, C. J. J. Paredis, and P. K. Khosla, "Heterogeneous teams of modular robots for mapping and exploration," *Autonomous Robots*, vol. 8, no. 3, pp. 293–308, 2000. DOI: 10.1023/a:1008933826411 [Online]. Available: https://link.springer.com/article/10.1023/A:1008933826411

[44] V. Kuo and R. Fitch, "Scalable multi-radio communication in modular robots," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 1034–1046, 2014. DOI: 10.1016/j.robot.2013.08.007

[45] A. Brunete Gonzalez, M. Hernando Gutierrez, E. Gambao Galan, and J. E. Torres Matesanz, "A behaviour-based control architecture for heterogeneous modular, multi-configurable, chained micro-robots," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1607–1624, Dec. 2012, ISSN: 0921-8890. DOI: 10.1016/j.robot.2012.09.019 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901200173X

[46] M.-K. Kvalsund, K. Glette, and F. Veenstra, "Centralized and decentralized control in modular robots and their effect on morphology," in *Artificial Life Conference Proceedings*, MIT Press, 2022, p. 49. DOI: 10.1162/isal_a_00533 [Online]. Available: https://direct.mit.edu/isal/article/doi/10.1162/isal_a_00533/112308/Centralized-and-Decentralized-Control-in-Modular

[47] U. Jahn, C. Wolff, and P. Schulz, "Concepts of a modular system architecture for distributed robotic systems," *Computers*, vol. 8, no. 1, p. 25, 2019. DOI: 10.3390/computers8010025 [Online]. Available: https://www.mdpi.com/2073-431X/8/1/25

[48] H. Bruyninckx, "OROCOS: Design and implementation of a robot control software framework," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:16080474

[49] M. Reichardt, T. Föhst, P. Fleischmann, M. Arndt, and K. Berns, "Principles in framework design applied in networked robotics," in *Proceedings of the 3rd IFAC Symposium on Telematics Applications (TA 2013)*, vol. 46, 2013, pp. 150–155. DOI: 10.3182/20131111-3-KR-2043.00016 [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1474667015343810

[50] O. Vermesan et al., "Internet of robotic things – converging sensing/actuating, hyperconnectivity, artificial intelligence and iot platforms," in *Cognitive Hyperconnected Digital Transformation: Internet of Things Intelligence Evolution*, O. Vermesan and J. Bacquet, Eds. River Publishers, 2017, ch. 4, pp. 97–155, ISBN: 9788793609105. DOI: 10.1201/9781003337584-4

[51] C. Lessi, A. Gavrielides, V. Solina, R. Qiu, L. Nicoletti, and D. Li, "5g and beyond 5g technologies enabling industry 5.0: Network applications for robotics," *Procedia Computer Science*, vol. 232, pp. 675–687, Jan. 2024. DOI: 10.1016/j.procs.2024.01.067

[52] S. Aoki, T. Yonezawa, and N. Kawaguchi, "Robotnest: Toward a viable testbed for iot-enabled environments and connected and autonomous robots," *IEEE Sensors Letters*, vol. 6, no. 2, pp. 1–4, 2022. DOI: 10.1109/LSENS.2021.3139624

[53] Connectivity Standards Alliance, "Zigbee specification," Connectivity Standards Alliance, Document 05-3474-23, Rev 23, 2023, Zigbee PRO 2023 (R23).

[54] A. Romero, C. Delgado, L. Zanzi, X. Li, and X. Costa-Pérez, "Online operation and orchestration of collaborative robots using 5g," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4356–4369, 2023. DOI: 10.1109/TNSM.2023.3281976

[55] A. Meseguer Valenzuela, J. Pérez, J. Silvestre-Blanes, and V. Payá, "Evaluation of combined network slicing and mec toward mobile robot applications in 5g networks," *IEEE Communications Standards Magazine*, vol. 10, pp. 280–286, Jan. 2025. DOI: 10.1109/MCOMSTD.2025.3633077

[56] G. Szabó, G. Seres, M. L. Mikecz, and T. Cinkler, "Assessment of the efficiency of 5g network exposure for the industrial internet of things," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2021, pp. 52–58. DOI: 10.1109/CSCN53733.2021.9686079

[57] P. Marti, M. Frigola, and M. Velasco, "Bandwidth management for distributed control of highly articulated robots," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 266–271. DOI: 10.1109/ROBOT.2005.1570130

[58] A. Romanov et al., "Modular reconfigurable robot distributed computing system for tracking multiple objects," *IEEE Systems Journal*, vol. 15, pp. 802–813, May 2020. DOI: 10.1109/JSYST.2020.2990921

[59] N. Li, H. Chen, M. Li, S. Yu, and Y. Huang, "Model predictive control: Past, present, and future," *Control Theory and Technology*, vol. 24, no. 2, pp. 173–193, 2026. DOI: https://doi.org/10.1007/s11768-026-00320-x [Online]. Available: https://www.sciengine.com/doi/10.1007/s11768-026-00320-x

[60] M. Alipourvarmazabadi and M. Ebner, "Robot localization: A comprehensive survey from classical methods to intelligent autonomy," *International Journal of Intelligent Robotics and Applications*, Mar. 2026. DOI: 10.1007/s41315-026-00528-9 [Online]. Available: https://link.springer.com/article/10.1007/s41315-026-00528-9

[61] S. Atiya and G. D. Hager, "Real-time vision-based robot localization," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 785–800, 1993, ISSN: 1042-296X. DOI: 10.1109/70.265922

[62] D. Yuen and B. MacDonald, "Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 217–226, 2005. DOI: 10.1109/TRO.2004.835452

[63] J. Bae, S. Lee, and J.-B. Song, "Use of coded infrared light for mobile robot localization," *Journal of Mechanical Science and Technology*, vol. 22, no. 7, pp. 1279–1286, 2008. DOI: 10.1007/s12206-008-0320-1

[64] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with rfid technology," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, May 2004, pp. 1015–1020. DOI: 10.1109/ROBOT.2004.1307283

[65] B. Sohn, J. Lee, H. Chae, and W. Yu, "Localization system for mobile robot using wireless communication with ir landmark," in *Proceedings of the 1st International Conference on Robot Communication and Coordination*, Jan. 2007, p. 6. DOI: 10.1145/1377868.1377876

[66] Y. Grigorovich, S. Geyman, I. Yusupov, A. Kharchevskii, I. Melchakova, and M. Udrov, "Long-range nfc device localization with millimeter-scale accuracy," *Photonics and Nanostructures - Fundamentals and Applications*, vol. 67, p. 101 476, 2025, ISSN: 1569-4410. DOI: 10.1016/j.photonics.2025.101476

[67] D. Jeong and K. Lee, "Distributed communication and localization algorithms for homogeneous robotic swarm," in *Distributed Autonomous Robotic Systems*, ser. Springer Tracts in Advanced Robotics, N.-Y. Chong and J.-B. Song, Eds., vol. 112, Springer, Tokyo, 2016, pp. 375–387. DOI: 10.1007/978-4-431-55879-8_28 [Online]. Available: https://link.springer.com/chapter/10.1007/978-4-431-55879-8_28

[68] J. Lwowski, A. Majumdat, P. Benavidez, J. J. Prevost, and M. Jamshidi, "Htc vive tracker: Accuracy for indoor localization," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 4, pp. 15–22, 2020. DOI: 10.1109/MSMC.2020.2969031

[69] E. M. Gorostiza, J. L. Lázaro Galilea, F. J. Meca Meca, D. Salido Monzú, F. Espinosa Zapata, and L. Pallarés Puerto, "Infrared sensor system for mobile-robot positioning in intelligent spaces," *Sensors*, vol. 11, no. 5, pp. 5416–5438, 2011, ISSN: 1424-8220. DOI: 10.3390/s110505416 [Online]. Available: https://www.mdpi.com/1424-8220/11/5/5416

[70] M. Menolotto, D.-S. Komaris, S. Tedesco, B. O'Flynn, and M. Walsh, "Motion capture technology in industrial applications: A systematic review," *Sensors*, vol. 20, no. 19, p. 5687, 2020. DOI: 10.3390/s20195687 [Online]. Available: https://www.mdpi.com/1424-8220/20/19/5687

[71] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002. DOI: 10.1109/TRA.2002.803462

[72] S. Funiak, M. Ashley-Rollman, P. Pillai, S. Goldstein, and J. Campbell, "Distributed localization of modular robot ensembles," in *Robotics: Science and Systems IV*. MIT Press, Jun. 2009, pp. 33–40, ISBN: 9780262258623. DOI: 10.7551/mitpress/8344.003.0009

[73] J. Zhu and S. S. Kia, "Cooperative localization under limited connectivity," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1523–1530, 2019. DOI: 10.1109/TRO.2019.2930404

[74] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, 2018. DOI: 10.1177/0278364918760698 [Online]. Available: https://doi.org/10.1177/0278364918760698

[75] L. Luft, T. Schubert, S. Roumeliotis, and W. Burgard, "Recursive decentralized collaborative localization for sparsely communicating robots," in *Robotics: Science and Systems 2016*, Jun. 2016. DOI: 10.15607/RSS.2016.XII.016

[76] S. S. Kia, J. Hechtbauer, D. Gogokhiya, and S. Martínez, "Server-assisted distributed cooperative localization over unreliable communication links," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1392–1399, 2018. DOI: 10.1109/TRO.2018.2830411

[77] N. Hao, F. He, Y. Yao, and Y. Hou, "Consistent batch fusion for decentralized multi-robot cooperative localization," *Control Theory and Technology*, vol. 22, pp. 638–651, Jul. 2024. DOI: 10.1007/s11768-024-00229-3

[78] L. Chen, C. Liang, S. Yuan, and L. Xie, "Relative localizability and localization for multirobot systems," *IEEE Transactions on Robotics*, vol. 41, pp. 2931–2947, 2025. DOI: 10.1109/TRO.2025.3544103

[79] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016. DOI: 10.1002/rob.21620 [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21620

[80] R. Yuan, B. Ji, Y. Gao, and H. Tao, "A review of lidar simultaneous localization and mapping techniques for multi-robot," *Robotica*, vol. 43, no. 9, pp. 3200–3240, Sep. 2025. DOI: 10.1017/S0263574725102178

[81] A. A. Alajami, N. Palau, S. Lopez-Soriano, and R. Pous, "A ros-based distributed multi-robot localization and orientation strategy for heterogeneous robots," *Intell. Serv. Robot.*, vol. 16, no. 2, pp. 177–193, Feb. 2023, ISSN: 1861-2776. DOI: 10.1007/s11370-023-00457-7 [Online]. Available: https://doi.org/10.1007/s11370-023-00457-7

[82] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017, ISBN: 9781107156302. [Online]. Available: https://cambridge.org

[83] C. Zhou, B. Huang, and P. Franti, "A review of motion planning algorithms for intelligent robots," *J. Intell. Manuf.*, vol. 33, no. 2, pp. 387–424, Feb. 2022, ISSN: 0956-5515. DOI: 10.1007/s10845-021-01867-z [Online]. Available: https://doi.org/10.1007/s10845-021-01867-z

[84] C. M. Clark and S. M. Rock, "Motion planning for multiple mobile robots using dynamic networks," in *Proceedings 2003 IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 532–538. DOI: 10.1109/ROBOT.2003.1242252

[85] Y. Guo and L. E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, IEEE, vol. 3, 2002, pp. 2626–2631. DOI: 10.1109/ROBOT.2002.1013625

[86] P. S. Bhat, J. Kuffner, S. C. Goldstein, and S. Srinivasa, "Hierarchical motion planning for self-reconfigurable modular robots," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 886–891. DOI: 10.1109/iros.2006.281742 [Online]. Available: http://www.cs.cmu.edu/~seth/papers/bhat06.pdf

[87] Y. Chen, U. Rosolia, and A. D. Ames, "Decentralized task and path planning for multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4337–4344, 2021. DOI: 10.1109/lra.2021.3068103 [Online]. Available: http://ames.caltech.edu/chen2021decentralized.pdf

[88] L. Antonyshyn, J. Silveira, S. Givigi, and J. Marshall, "Multiple mobile robot task and motion planning: A survey," *ACM Comput. Surv.*, vol. 55, no. 10, Feb. 2023, ISSN: 0360-0300. DOI: 10.1145/3564696 [Online]. Available: https://doi.org/10.1145/3564696

[89] V. Vonasek, M. Saska, K. Kosnar, and L. Preucil, "Global motion planning for modular robots with local motion primitives," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2465–2470. DOI: 10.1109/ICRA.2013.6630912 [Online]. Available: https://ieeexplore.ieee.org/document/6630912/

[90] A. Pamecha, I. Ebert-Uphoff, and G. Chirikjian, "Useful metrics for modular robot motion planning," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 531–545, 1997. DOI: 10.1109/70.611311

[91] T. Larkworthy and S. Ramamoorthy, "An efficient algorithm for self-reconfiguration planning in a modular robot," in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 5139–5146. DOI: 10.1109/ROBOT.2010.5509482

[92] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Implan: Scalable incremental motion planning for multi-robot systems," in *Proceedings of the 7th International Conference on Cyber-Physical Systems*, ser. ICCPS '16, Vienna, Austria: IEEE Press, 2016. DOI: 10.1109/iccps.2016.7479105

[93] A. Majed, H. Harb, A. Nasser, and B. Clement, "Run: A robust cluster-based planning for fast self-reconfigurable modular robotic systems," *Intell. Serv. Robot.*, vol. 16, no. 1, pp. 75–85, Jan. 2023, ISSN: 1861-2776. DOI: 10.1007/s11370-023-00454-w [Online]. Available: https://doi.org/10.1007/s11370-023-00454-w

[94] J. Seo, J. Paik, and M. Yim, "Modular reconfigurable robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 63–88, 2019. DOI: 10.1146/annurev-control-053018-023834 [Online]. Available: https://doi.org/10.1146/annurev-control-053018-023834

[95] D. Saldaña et al., "A decentralized algorithm for assembling structures with modular robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2017, pp. 2736–2743. DOI: 10.1109/IROS.2017.8206101

[96] C. Liu, J. Liu, R. Moreno, F. Veenstra, and A. Faina, "The impact of module morphologies on modular robots," in *2017 18th International Conference on Advanced Robotics (ICAR)*, 2017, pp. 237–243. DOI: 10.1109/ICAR.2017.8023524

[97] S. Chennareddy, A. Agrawal, and A. Karuppiah, "Modular self-reconfigurable robotic systems: A survey on hardware architectures," *Journal of Robotics*, vol. 2017, p. 5 013 532, 2017. DOI: 10.1155/2017/5013532

[98] S. Hauser, M. Mutlu,
bibinitperiod Leziart, H. Khodr, A. Bernardino, and
bibinitperiod Ijspeert, "Roombots extended: Challenges in the next generation of self-reconfigurable modular robots and their application in adaptive and assistive furniture," English, *Robotics and Autonomous Systems*, vol. 127, May 2020, Publisher Copyright: © 2020 Elsevier B.V., ISSN: 0921-8890. DOI: 10.1016/j.robot.2020.103467

[99] H. Kawano, "Tunneling-based self-reconfiguration of heterogeneous sliding cube-shaped modular robots in environments with obstacles," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 825–832. DOI: 10.1109/ICRA.2017.7989100

[100] B. Piranda and J. Bourgeois, "A distributed algorithm for reconfiguration of lattice-based modular self-reconfigurable robots," in *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, Heraklion, Crete, Greece: IEEE, 2016, pp. 1–9. DOI: 10.1109/PDP.2016.40

[101] A. M. Romanov, M. P. Romanov, and E. I. Shestakov, "A novel architecture for control systems of modular reconfigurable robots," in *2017 IEEE II International Conference on Control in Technical Systems (CTS)*, 2017, pp. 131–134. DOI: 10.1109/CTSYS.2017.8109507

[102] B. S. I. Group, *Bluetooth technology website*, Accessed: 2026-07-21, 2019. [Online]. Available: https://www.bluetooth.com/

[103] R. Fitch and R. Lal, "Experiments with a zigbee wireless communication system for self-reconfiguring modular robots," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ser. ICRA'09, Kobe, Japan: IEEE Press, 2009, pp. 3965–3970, ISBN: 9781424427888. DOI: 10.1109/robot.2009.5152807

[104] W. Qiu, P. Hao, and R. J. Evans, "An efficient self-healing process for zigbee sensor networks," in *2007 International Symposium on Communications and Information Technologies*, 2007, pp. 1389–1394. DOI: 10.1109/ISCIT.2007.4392233

[105] U. P. Schultz, "Distributed control diffusion: Towards a flexible programming paradigm for modular robots," in *Proceedings of the 1st International Conference on Robot Communication and Coordination*, ser. RoboComm '07, Athens, Greece: IEEE Press, 2007, ISBN: 9789639799080. DOI: 10.5555/1377868.1377887

[106] Z. Butler, R. Fitch, and D. Rus, "Experiments in distributed control for modular robots," in *Algorithmic Foundations of Robotics V*, ser. Springer Tracts in Advanced Robotics, vol. 7, Springer Berlin Heidelberg, 2003, pp. 407–422. DOI: 10.1007/3-540-36268-1_27 [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-36268-1_27

[107] C. Zhang, L. Zhou, and Y. Li, "Pareto optimal reconfiguration planning and distributed parallel motion control of mobile modular robots," *IEEE Transactions on Industrial Electronics*, vol. 71, pp. 9255–9264, Jan. 2023. DOI: 10.1109/TIE.2023.3321997

[108] A. Turetta, G. Casalino, and A. Sorbara, "Distributed control architecture for self-reconfigurable manipulators," *The International Journal of Robotics Research*, vol. 27, pp. 481–504, Mar. 2008. DOI: 10.1177/0278364907085565

[109] S. Dehnavi, M. Koedam, A. Nelson, D. Goswami, and K. Goossens, "Compros: A composable ros2 based architecture for real-time embedded robotic development," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic: IEEE Press, 2021, pp. 6449–6455. DOI: 10.1109/IROS51168.2021.9636590 [Online]. Available: https://doi.org/10.1109/IROS51168.2021.9636590