REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)
12-02-2017

2. REPORT TYPE
Final Report

3. DATES COVERED (From - To)
18-Mar-2011 - 30-Sep-2016

4. TITLE AND SUBTITLE
Final Report: Mid-Level Planning and Control for Articulated Locomoting Systems

5a. CONTRACT NUMBER
W911NF-11-1-0061

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0620BK

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS
Howie Choset

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Carnegie Mellon University
5000 Forbes Avenue

Pittsburgh, PA 15213 -3589

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
59651-MS-DRP.7

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
We changed directions with the proposed work from midlevel control of articulated systems to design and control of series elastics actuators for snake robots. The ability to adapt to and locomote through unstructured terrain remains a key weakness in the abilities of all snake robots. This problem partly stems from the fact that the gaits that we use to locomote are shape-driven (more accurately discrete-curvature driven), and our locomotion is primarily through rolling ground contact. While the shape of the snake can deform significantly to adapt to an environment, our ability to locomote typically depends exploiting structure in the environment through careful

15. SUBJECT TERMS
series elastic actuators, torque-based control, snake robot

16. SECURITY CLASSIFICATION OF:
a. REPORT UU
b. ABSTRACT UU
c. THIS PAGE UU

17. LIMITATION OF ABSTRACT
UU

15. NUMBER OF PAGES

19a. NAME OF RESPONSIBLE PERSON
Howard Choset

19b. TELEPHONE NUMBER
412-268-4985

Report Title

Final Report: Mid-Level Planning and Control for Articulated Locomoting Systems

ABSTRACT

We changed directions with the proposed work from midlevel control of articulated systems to design and control of series elastics actuators for snake robots. The ability to adapt to and locomote through unstructured terrain remains a key weakness in the abilities of all snake robots. This problem partly stems from the fact that the gaits that we use to locomote are shape-driven (more accurately discrete-curvature driven), and our locomotion is primarily through rolling ground contact. While the shape of the snake can deform significantly to adapt to an environment, our ability to locomote typically depends exploiting structure in the environment through careful choices of the snake's shape.

Force control is potentially a good perspective from which to approach the problem of adaptive control and locomotion. One could argue that all of the snake robot's locomotion is a result of force interactions, rather than shape interactions. A key development from the MIT Leg Lab to address this problem has been the series elastic actuator. By essentially low-passing the output of the actuator, a series elastic actuator provides more accurate and stable force control and protects the drive components. It also allows practical force sensing using common position-control sensors, such as potentiometers and encoders.

We propose implementing a series elastic actuator in the context of our existing snake robots. Specifically, we have built rotational elasticity into to the joints of each snake robot module, and used the rotational displacement of the elastic member to infer the torque exerted by the joint. We have begun developing controllers for this new system.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL: 1

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
08/03/2014	3.00	David Rollinson, Steven Ford, Ben Brown, Howie Choset. Design and Modeling of a Series Elastic Elementfor Snake Robots , Proceedings of the 6th Annual Dynamic Systems andControl Conference . 23-OCT-13, . : ,
08/31/2015	5.00	David Rollinson, Yigit Bilgen, Ben Brown, Florian Enner, Steven Ford, Curtis Layton,, Justine Rembisz, Mike Schwerin, Andrew Willig, Pras Velagapudi, Howie Choset. Design and Architecture of a Series Elastic Snake Robot, Intl Conf on Intellient Robot Systems. 15-SEP-14, . : ,
08/31/2015	6.00	David Rollinson, Kalyan Vasudev Alwala, Nico Zevallos, Howie Choset. Torque Control Strategies for Snake Robots, Intl Conf on Intellegient Robot Systems. 15-SEP-14, . : ,
09/16/2012	1.00	Florian Enner, , David Rollinson, Howie Choset. Simplified Motion Modeling for Snake Robots, IEEE Conference on Robotics and Automation. 11-MAY-12, . : ,
09/16/2012	2.00	Cornell Wright , Austin Buchan , Ben Brown , Jason Geist, Michael Schwerin, David Rollinson , Matthew Tesch, Howie Choset. Design and Architecture of the Unified Modular Snake Robot, IEEE Conference on Robtoics and Automation. 12-MAY-12, . : ,

TOTAL: **5**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper

TOTAL:

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Systems and Methods for Modular Units in Electro-Mechanical Systems
Series Elastic Actuators for Robot and Robot Systems

Patents Awarded

Systems and Methods for Modular Units in Electro-Mechanical Systems

Awards

Graduate Students

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Howie Choset	0.08	
FTE Equivalent:	**0.08**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME
Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
Matt Tesch	0.50
Dave Rollinson	0.50
Florian Enner	1.00
Curtis Layton	0.50
FTE Equivalent:	**2.50**
Total Number:	**4**

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

The bulk of the effort was advancing the usability of the modules as well as designing a newer easier-to-assemble one. We also created a set of educational modules that use the modules in a standard kinematics class.

Technology Transfer

THe PI, along with his students, started a company called Hebi Robotics that is commericializing the modular series elastic actuators created in this effort

Final Report

Mid-Level Planning and Control for Articulated Locomoting Systems

Contract # W911NF1110061

Design and Architecture of a Series Elastic Snake Robot

Like previous generations of our robots, the *SEA Snake* consists of a number of identical 1-DOF modules in which the actuated axes are oriented in the lateral and dorsal planes of the robot. Most notably, this latest generation contains a series elastic actuator in each module. This enables compliant motion over rough terrain and fine torque control on each joint. Additionally, each module contains a 32-bit processor and 100 Mbps Ethernet data bus, a 400X improvement in bandwidth over the RS-485 serial communications of our previous robot, the *Unified Snake* [102]. The modules also have greater torque output, improved sealing, and a rugged tool-free interface. Table B.1 presents the specifications of the *SEA Snake* robot.

B.1 Mechanical Overview

Each module possesses a self-contained 1-DOF joint, allowing for a full 180° of rotation. Modules are interfaced together and alternately aligned in accordance with the robot's lateral and dorsal planes. A typical robot consists of 16 modules linked together, with

Table B.1: Overview of *SEA Snake* specifications.

Dimensions	Diameter: 5.1 cm Length (module): 6.4 cm Length (full 16 module robot): 1.174 m
Mass	Module: 205 g Full 16 module robot: 3.657 kg
Actuation	Max Torque: 7 N-m Max Speed: 33 RPM
Power	48 V Current (resting): 40 mA Current (max): 600 mA
Communication	100 Mbps Ethernet
Sensing	Angular Position and Velocity Output Torque 3-axis Accelerometer 3-axis Gyro Temperature Voltage Current

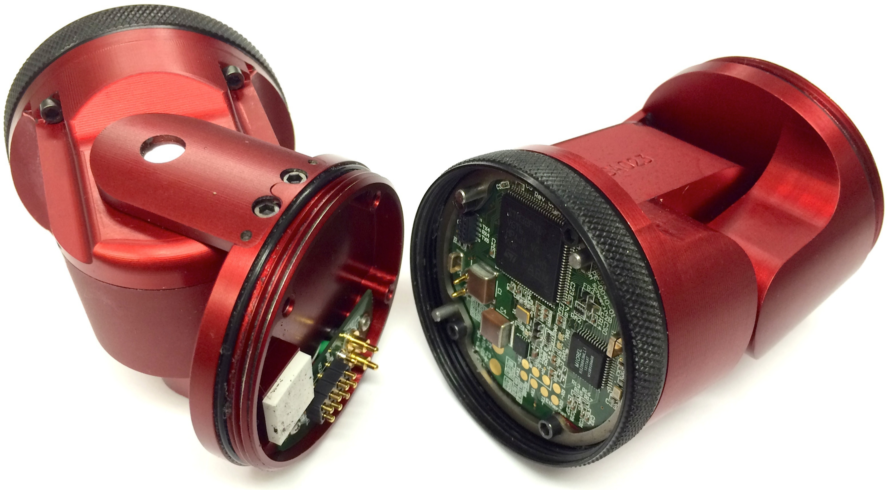

Figure B.1: Photo of a *SEA Snake* Module. The module is 5 cm (2 in.) in diameter and provides a 1-DOF rotary motion of +/- 90°.

unique head and tail modules.

A driving design requirement of the *SEA Snake* was ease of customization. In addition to the 16 rotary DOF design, modules can be added, removed, interchanged, or replaced with novel mechanisms. Any device meeting the interface requirements can be included in the chain.

B.1.1 Motor-Geartrain

The *SEA Snake* modules are driven by a modified Maxon EC 20 flat motor with a nominal speed of 9300 RPM. The steel pinion gear on the motor's output shaft transfers rotation through a geartrain containing 3 steel and brass compound gears. The cumulative gear ratio is 349:1 to create high-torque joints. This motor and geartrain combination provides a maximum output torque of 7 N-m and a maximum speed of 33 RPM.

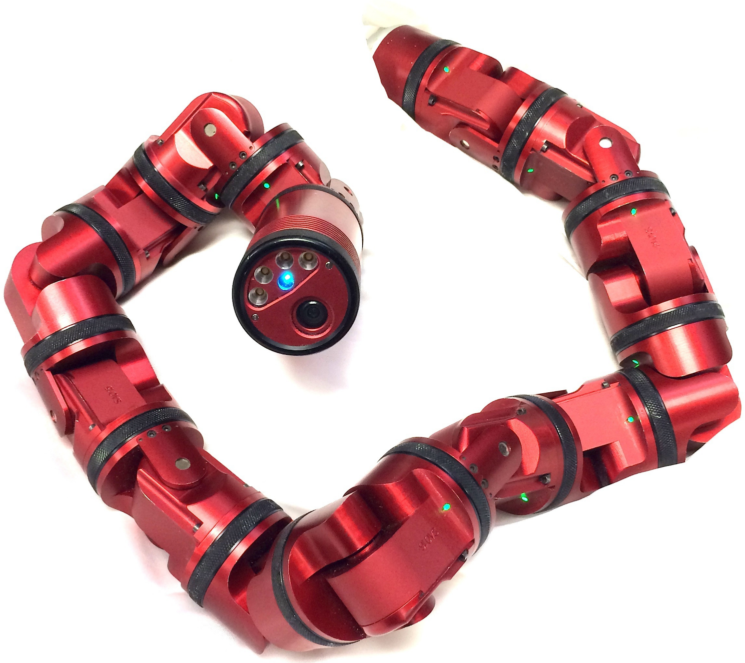

Figure B.2: Photo of the Series Elastic Actuated Snake Robot (*SEA Snake*).

B.1.2 Sealed Housing

The housing of each module is machined from 7075 aluminum and anodized red to prevent wear and corrosion. Components are densely assembled inside to minimize volume, as illustrated in the module cross-section in Fig. B.3. O-rings laid in machined grooves seal the module at each interface. The robot meets IP66 standards, meaning it is splash-proof. Future iterations will aim for IP68, or water submersible. Additionally, an effort was made to minimize external fasteners in the design. The previous snake robot, the *Unified Snake*, has 14 external fasteners per module, while the *SEA Snake* modules have only 4.

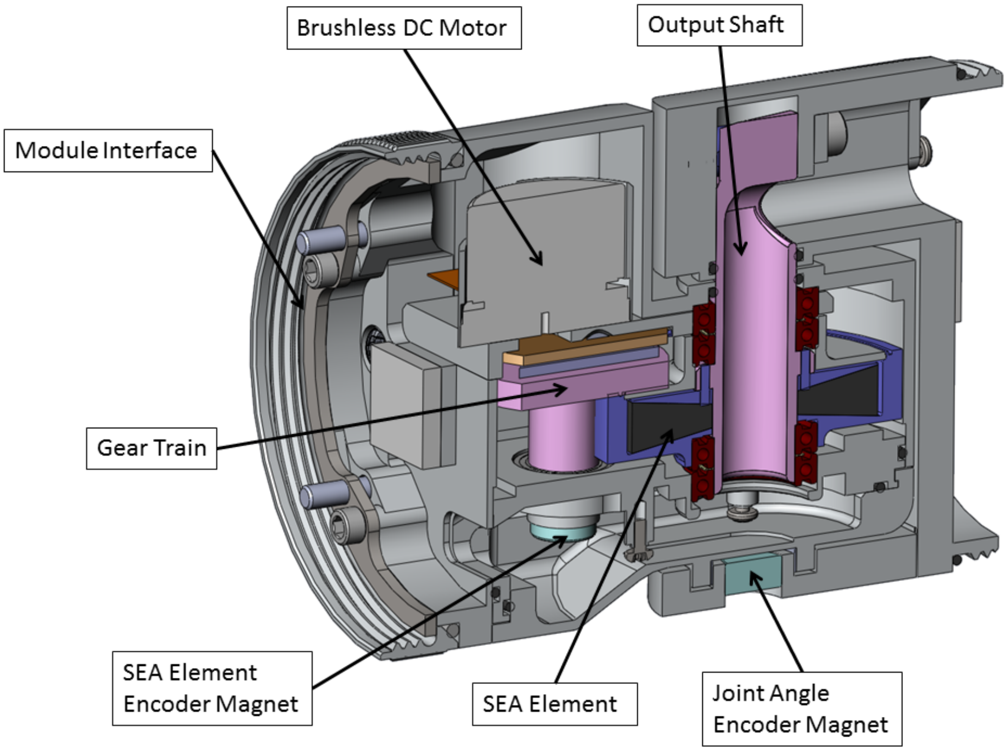

Figure B.3: CAD model cross-section of a *SEA Snake* module.

B.1.3 Mechanical Interface

The intermodular interface features a rugged, tool-less design. Modules are aligned with dowel pins and matching recesses. A freely-spinning threaded collar, held in place by a retaining ring, is turned by hand to lock adjacent modules together. An electrical connection is made between two modules with spring-pin connectors on the interface board touching target areas on the control board.

O-rings seal the collar at both ends. Modules can be connected and disconnected quickly and repeatedly. The connections are secure and resist shock and stress. Any device with matching threads, 48V and Ethernet compatibility can be interfaced with a module, allowing for freedom of design and customization.

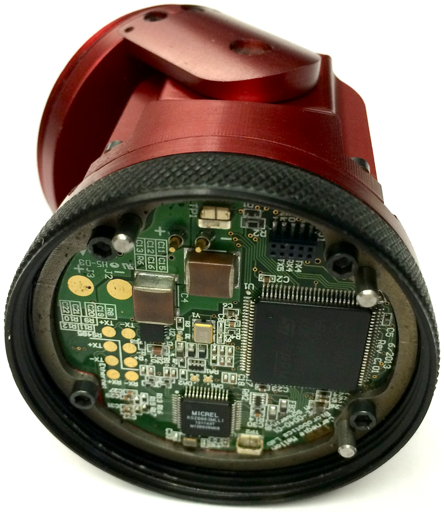

Figure B.4: Photo of the modular *SEA Snake* interface. Dowel pins provide alignment while the threaded collar mechanical secures and seals the modules. The electrical connection is made with spring-pin connectors touching target areas on the control board.

B.1.4 Series Elasticity

The *SEA Snake* features series elastic actuators. A rubber elastomer bonded between two rigid plates is torsionally sheared during actuation [80]. The elastomer's tapered conical cross section shown in Fig. B.5 is similar to the constant-shear-stress design introduced in [80]. The spring here is different in that it is molded directly to the output gear of our geartrain. The rigid plate on top is then attached to the output shaft of the system through a number of pins. Another plate is swaged onto this assembly in order to keep the output gear and output shaft aligned with the rest of the geartrain.

The elastomer is molded from Natural Rubber of Shore A Durometer 50. As a torsional spring, its stiffness is roughly characterized by a spring constant of 12 N-m/rad and a maximum rotational deflection of approximately 0.6 radians. Our research is currently exploring ways to estimate the output torque from the elastomer deflection and how to calibrate its parameters online [19].

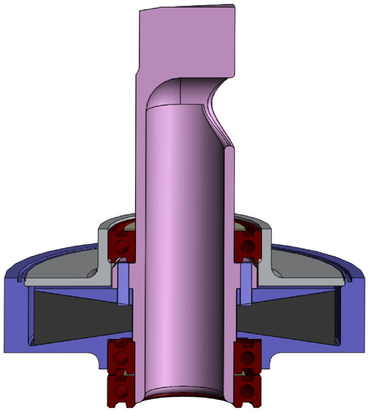

Figure B.5: CAD model cross-section of a module's output shaft assembly. The rubber is bonded to the tapered surfaces of the output gear and top washer. The output shaft is hollow, allowing wires to pass through the center of rotation of the module.

B.1.5 Head and Tail Modules

Both a head and tail module were designed utilizing the custom modular interface and they are pictured in Fig. B.6. In addition to providing their specialized functions, these modules demonstrate the potential for the use of other modules that could easily be integrated into the *SEA Snake* system.

The head module includes a high-definition camera to provide the user with a live video feed while the four LEDs are available for illuminating darker environments. The head module housing is designed with fins to increase surface area and improve heat transfer from the electronics to the surrounding environment. The LEDs are protected by an o-ring sealed acrylic window while the camera's lens is protected by an o-ring sealed sapphire glass window.

In order to connect a tether to the snake, we custom-designed a tether connector that is sealed and load-bearing. The connector uses keys and keyways to provide a quick and very easy blind connection. Additionally, there are spring-loaded pins within the connector which ensures that the connector's housing bears all of the load. The tail has this connector as well as a slipring integrated within its design.

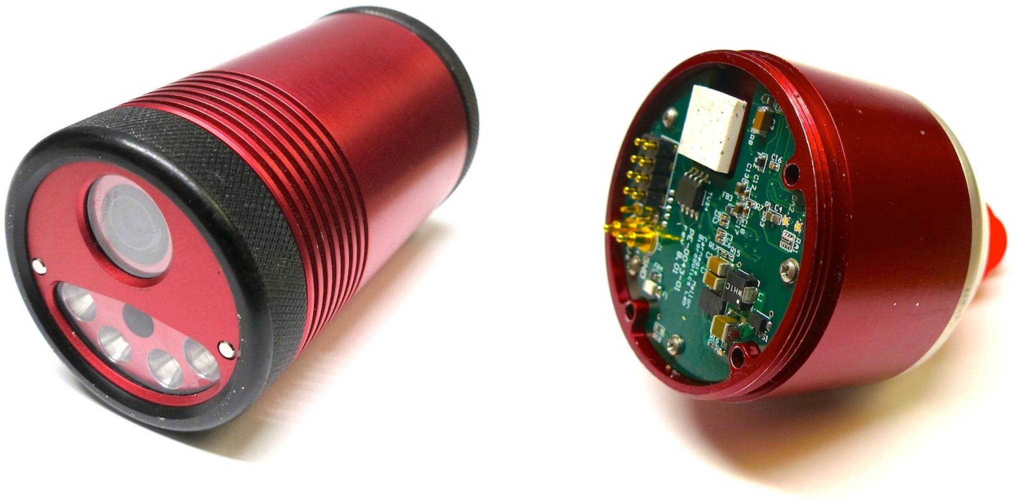

Figure B.6: Photos of *SEA Snake* head and tail modules.

B.2 Electronics Overview

While the electronics in the previous generation robot were robust, we set out to design the *SEA Snake* electronics with a focus on ease of development to facilitate future research, such as more advanced control algorithms and onboard sensor fusion. To this end, we moved from an 8-bit AVR processor to a 32-bit ARM Cortex M4F running at 7X the clock speed. We also wanted to enable higher frequency external control and sensor feedback and to move from analog to digital video. To satisfy these increased data requirements we moved from a 250kbps RS-485 bus to a switched 100Mbps Ethernet network.

B.2.1 Communication

One hindrance to adopting Ethernet in embedded devices is the size of the isolation transformers typically required. Our solution was to integrate a three port Ethernet switch into each module. This keeps the wire length down to a few centimeters, which is far shorter than the electrical propagation distance during a single 10ns bit period.

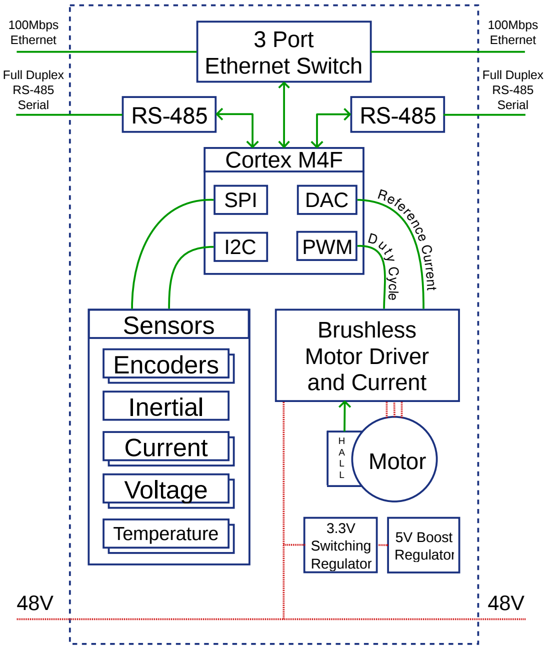

Figure B.7: Module Electronics Block Diagram.

This allowed us to relax the controlled impedance requirements, and to replace the transformer isolation with capacitive isolation, which has a far smaller PCB footprint. We do provide transformer isolation in the tail module to insure signal integrity over the tether.

In addition to the Ethernet link each module has a full duplex differential serial link to each of its immediate neighbors for future uses including 'autonumbering' of the modules. We have implemented a system in which the modules communicate their IP address to their immediate neighbors and the robot is able to self-discover the order in which the modules are connected. This greatly aids in field maintenance and flexibility of the robot as a research platform.

B.2.2 Electrical Interface

On the previous snake robot design the module interface was along the axis of rotation, requiring us to create a custom connector to fit around the coaxial magnet used for position feedback. The *SEA Snake* design moves the module interface away from the axis of rotation which has many advantages for the electrical interconnection. Because we have more space for the interconnection we use spring contacts that mate with gold plated PCB pads instead of very thin pins. This arrangement is much more tolerant to misalignment, eliminating the problem of bent or broken pins. The new design also moves the intermodule wiring terminations as far as possible from twisting motion caused by joint rotation, greatly reducing the risk of fatigue failures at the crimp/solder connections.

B.2.3 Motor Control

The *SEA Snake* is our first generation to include a brushless motor. We drive the motor with an integrated single chip solution to which we add current sensing. The series elastic member also required the addition of a second magnetic rotation encoder which allows us to measure the deflection. We use a 48V system voltage and 36V motor combined with thermal modeling to allow the system to safely exceed the continuous operation ratings of the motor for brief periods.

B.2.4 Sensors

Every generation of our snake robot has benefited from rapid development in inertial sensor technology driven by consumer electronics, with the current generation featuring a single chip solution with 3-axis gyro, 3-axis accelerometer, and 3-axis magnetometer which costs 1/3 the price of the 2-axis gyro from early in the previous generation. We retained all the other sensing present in the previous generation including voltage and temperature monitoring, but we also added a current sensor to measure the consumption of the entire module in addition to the motor current measuring circuit. This generation also adds externally visible PWM controlled RGB LEDs without sacrificing environmental sealing to provide immediate feedback on module status.

B.2.5 Camera Head

The use of Ethernet as the standard communication interface between modules enables the use of readily available, IP security cameras as the basis for the head module of the snake. In the Unified Snake architecture, the video was transmitted over a dedicated, analog bus that supported only standard definition video. By moving to an Ethernet-based camera, the video stream is piped over the same set of cabling as the motion control commands. In addition, the bandwidth that is needed to transmit a high definition video stream is a fraction of the total bandwidth available on the system, paving the way for the use of multiple cameras and additional sensors.

The camera module that has been designed to be used for the *SEA Snake* is primarily built around the circuit board and camera that were harvested from a commercially available security camera. A custom circuit board was designed to provide power to the camera, to support additional sensors such as an IMU and a pressure sensor, as well as to control high-brightness LEDs for camera illumination. The camera and electronics were packaged into a simple aluminum tube featuring the same standard mechanical and electrical interface as the regular snake modules.

B.3 Firmware Overview

The *SEA Snake* robot is a research platform and is constantly reconfigured to support new physical configurations, module types, and sensor types. In addition, researchers with a range of experience must be able to use the platform for research in controls, perception, and planning. To support these research goals, we developed a modular firmware that provides an API supporting both high-level and low-level control abstractions.

B.3.1 OS and Hardware Abstraction Layer

The core of the firmware is an RTOS that separates the setup and upkeep of various hardware modules into separate threads. ChibiOS/RT[1] was selected due to its out-of-the-box support for STM32 Cortex processors and a hardware abstraction layer supporting peripherals such as Ethernet (via the lwIP[2] stack), serial, analog-digital conversion, and pulse-width modulation. Each functional component of the module is wrapped in a C++ class abstracting its setup and configuration into a statically-initialized singleton. This wraps over the hardware abstraction layer provided by ChibiOS, combining multiple hardware peripherals blocks related to each component. For example, we internally initializes a serial interface, digital IO lines, and thread-safety, but exposes only simple read and update functions. This allows non-technical users to safely reconfigure hardware for specific purposes.

B.3.2 Communication

Modules communicate with each other and client software via Google Protocol Buffer[3] messages. Protocol Buffers define a fixed serialization format for typed data structures that is supported across multiple computing platforms. We ported the Google C++ library for Protocol Buffers to ChibiOS to allow messages to be encoded and decoded on the SEASnake module.

Rather than providing remote-procedure calls (RPC), we define a single "meta"-message which defines a number of optional fields that encode module parameters and data streams. Any computing environment with Protocol Buffer support can then

[1] `http://www.chibios.org/`

[2] `http://savannah.nongnu.org/projects/lwip/`

[3] `https://code.google.com/p/protobuf/`

interact with the snake by sending and receiving streams of these meta-messages. An advantage of this protocol is that it is fundamentally stateless and messages can be handled transactionally, naturally supporting multiple concurrent asynchronous connections.

This protocol is exposed identically over TCP sockets, UDP sockets, and the serial interface between modules, allowing a variety of flexible communication strategies to be used in various situations, depending on available bandwidth, required level of control, and interface hardware. Modules can propagate local information (such as topology) by exchanging data over the serial interface to neighboring modules, while inexpensive "dumb" components which contain only low-speed microcontrollers can be bridged to Ethernet via neighboring modules to implement low-cost special-purpose functions.

B.3.3 Motion Control

Modules support angular position, velocity, and torque control through cascaded PID control, as shown in Fig. 8.6. Each PID controller runs at 1kHz, although target setpoints may be updated less frequently, typically at 100 - 200 Hz. Independent position and velocity outer loops generate torque commands, which are combined with a desired feed-forward torque to define a setpoint for *output* torque which is maintained by the inner torque controller.

The inner torque controller is able to directly compare desired and actual output torque by directly observing spring deflection as the difference between the two encoder positions. This error is used to compute a PWM command to the motor which applies appropriate torque to the input of the spring and gear train.

In the proportional controller, additional features were added to help compensate for common geartrain nonlinearities, as illustrated in Fig. B.8. To prevent oscillations

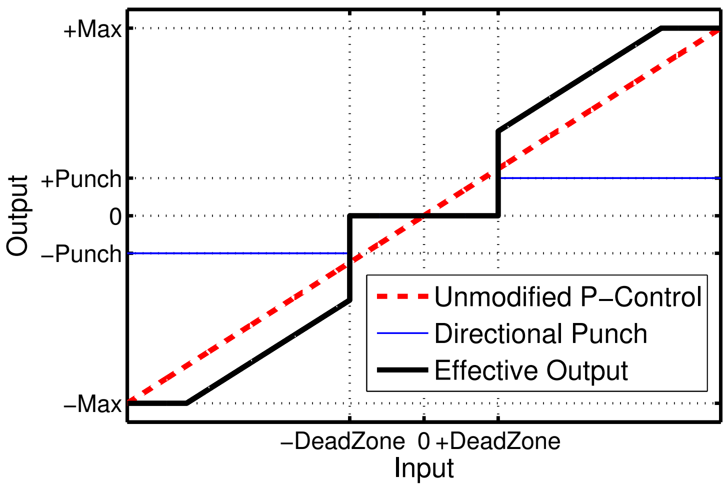

Figure B.8: The modified proportional controller, showing the effect of the deadzone and punch parameters on the output.

due to gear blacklash and sensor noise, we added a 'deadzone' within which errors are assumed to be zero. To overcome geartrain stiction, a 'punch' factor (directional offset represented by the blue dotted line) was added to help with stiction in the gear train. Finally maximum output limits are also set for each controller. The dashed red line in Fig. B.8 shows the output of a theoretical proportional controller, the black solid line shows the output of our actual implementation. To mitigate windup of the integral term we limit its output to the difference between the PD output and the a set output level [67]. For example, using this method, if PD control already reaches this level the integral term is reduced to 0.

B.3.4 Thermal Modeling

An important challenge for mobile robots is to safely extract as much performance as possible out of their actuators. For brushless motors, the primary limitation on performance is heat buildup in the motor windings [95]. We use online estimation of each module's motor winding temperature to fully exploit the motor's performance envelope beyond the continuous duty ratings. Figure B.10 shows a plot of the power

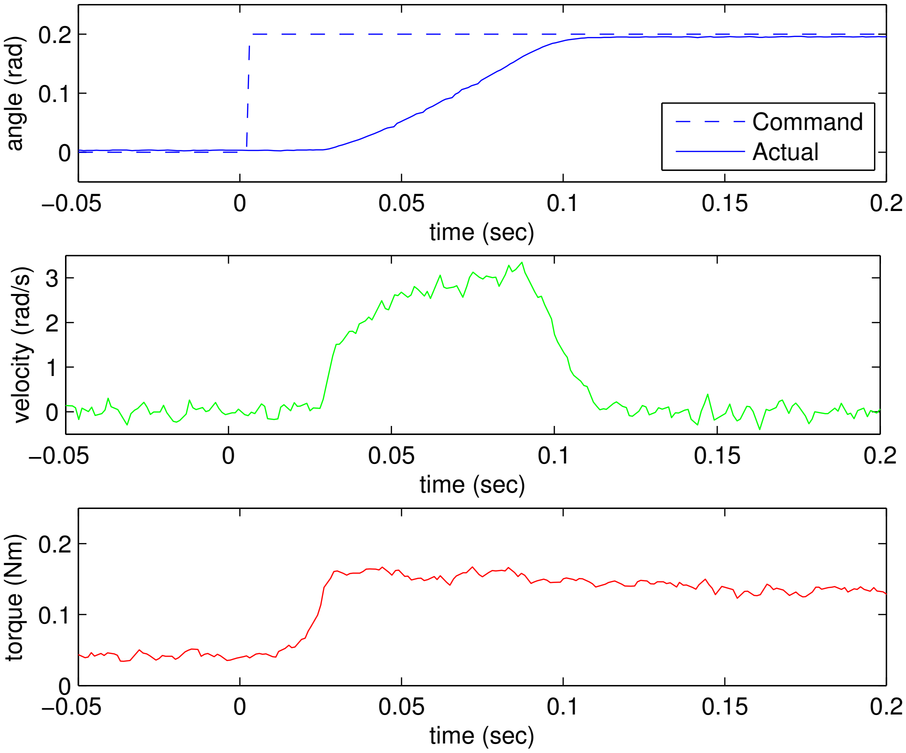

Figure B.9: The response of the angular position (top), velocity (middle), and torque (bottom) to a step input in position. This was performed on an unloaded module, and as such the torque readings mostly reflect the static friction of the module's shaft seals.

dissipation and estimated temperature of the motor windings while the motor is repeatedly stalled. The estimated winding temperature is based on a temperature sensor near the motor, the sensed current draw of the motor, and a model of the internal thermal resistances and capacitances of the motor similar to the method presented in [94].

B.4 Conclusion and Future Work

The *SEA Snake* currently consists of a series of extremely capable 1-DOF modules. However, there are a number of avenues of future work. These consist of ongoing improvements to the existing modules, mostly in firmware, and the development of different modules that share the electrical, mechanical, and software interfaces detailed

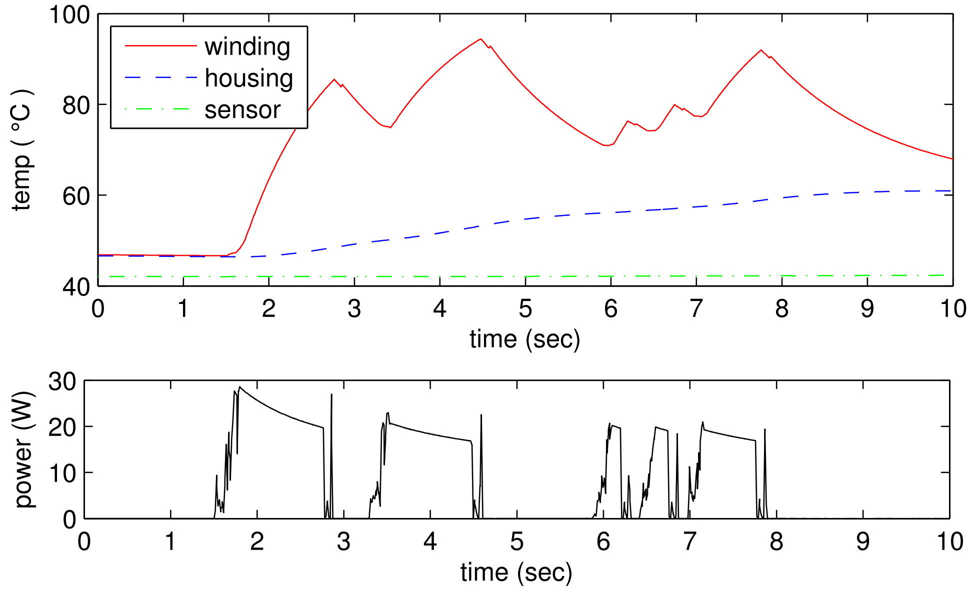

Figure B.10: Top: Estimated temperature of the motor windings in a *SEA Snake* module, based on the thermal model. Bottom: Dissipated motor power based on the measured current draw and the resistance of the motor's windings.

above.

The most exciting avenue for future work will be the creation of a number of new modules for the snake robot, based on the modular interface. Possible future modules include different head modules with cameras and other exteroceptive sensing, battery and wireless communication modules for tetherless operation, tracked or wheeled modules for improved mobility in rough terrain, and 'hockey puck' modules that expose general purpose inputs and outputs to easily test external sensors like force sensing skins.

We feel the architecture we have developed is general-purpose enough that topologies other than a snake may be feasible. For example, connecting a number of these modules together to a central chassis would enable the rapid prototyping of a field-ready legged robot with torque and position control on each of its joints.

Robust State Estimation

To help provide better feedback and improve an operator's situational awareness, we have integrated MEMS accelerometers and gyros into each module of our snake robots. Prior work from our group has already used an extended Kalman filter (EKF) to fuse these distributed sensors and achieve an estimate of the robot's pose [76]. By using knowledge of the robot's cyclic controller (gait) and taking advantage of an averaged body frame that we call the *virtual chassis*, we have been able to estimate a snake robot's orientation, even when it undergoes highly dynamic motions.

Unfortunately, our previous work has limitations in terms of its robustness in real-world field use. Frequent communication dropouts or corrupted data from the modules would sometimes cause the EKF to diverge. Additionally, the need for the state estimator to have explicit knowledge of the robot's gait equation means that it has to be tightly integrated with the gait framework that we use for control. This work addresses these issues with two contributions. First, we formulate the state estimation problem in a way that leverages redundancies in the proprioceptive information provided by the robot's joint angle encoders and inertial sensors. In particular, we are able to redundantly estimate the robot's kinematic state, the angles, angular velocities, and angular accelerations of the robot's joints) by using the inertial sensors in each module

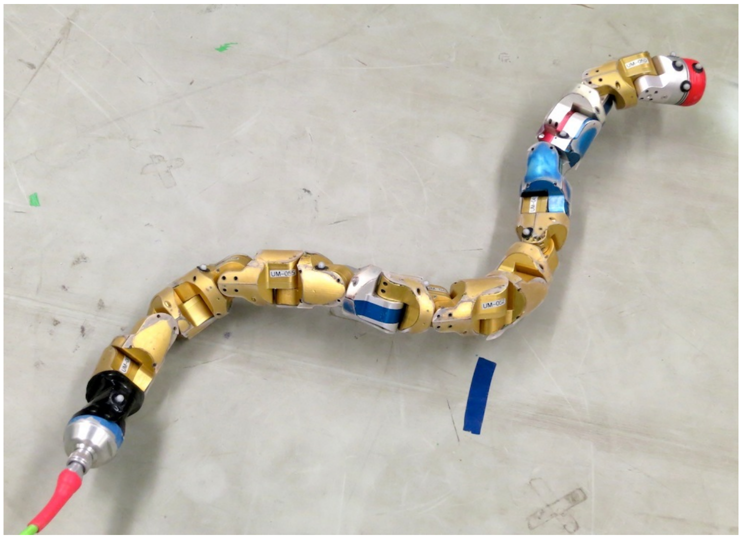

Figure 4.1: The *Unified Snake* robot, with markers attached for ground-truth motion capture.

to complement the readings from each module's joint angle encoders. Second, we introduce a novel outlier detector that can identify corrupted measurement data with a minimal amount of tuning.

The methods and results in this thesis are an expansion of previous preliminary results in state estimation [79]. Previous work assumed a second-order process and measurement models and static process and measurement noise. In order to better model the sensor data from the robot, this work moves to third-order kinematic models with noise models that are adjusted dynamically based on the estimated state. This work also examines how the choice of body frame affects the accuracy of state estimation and presents an alternative method for calculating an averaged *virtual chassis* body frame. New results are presented for these improvements, and we provide analysis of the practical benefits of distributed redundant sensing for robots.

4.1 Choice of Body Frame

In previous work [75, 76] we have demonstrated the benefits of using an averaged body frame that we call the *virtual chassis*. The virtual chassis is a body frame that is aligned

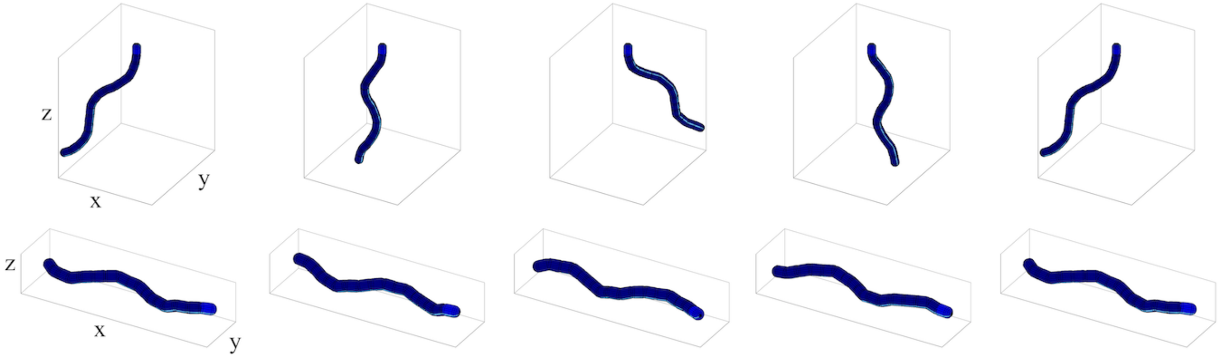

Figure 4.2: A montage of the robot in sidewinding, shown in five different positions spaced evenly throughout one complete gait cycle. The top row of images show the pose of the robot in a body frame fixed to the head link. The bottom row shows the pose corresponding to the virtual chassis body fame at the same points in the gait cycle.

with the principle components of the robot's overall shape, as shown in Fig. 4.2. The calculation of this body frame is performed as part of the measurement model at every iteration of the filter, using an SVD to identify the principle components of the robot's shape [76]. Details on the calculation of this body frame are presented in Appendix A.

The virtual chassis body frame has the advantage that it approximately separates the robot's internal shape changes from its external motions in the world, enabling more accurate and stable state estimation with generic constant velocity process models. Additionally, the state of the snake robot in this body frame is more intuitive to the operator, since the notions of up-down and left-right are aligned with the overall shape of the robot (Fig. 4.3). We have leveraged this overall shape alignment of the virtual chassis to infer ground contact for simple kinematic motion models [16, 17], and we rely on it heavily as a visualization tool for designing new gaits and motions.

Results are presented in Section 4.5 on the effect that the choice body frame has on the accuracy of state estimation. We investigate four different choices of body frame. In addition to two different formulations of the virtual chassis, we use a body frame fixed to the head module of the robot, and a body frame where the origin is at the geometric center of the robot (as is done in the virtual chassis) but the orientation is

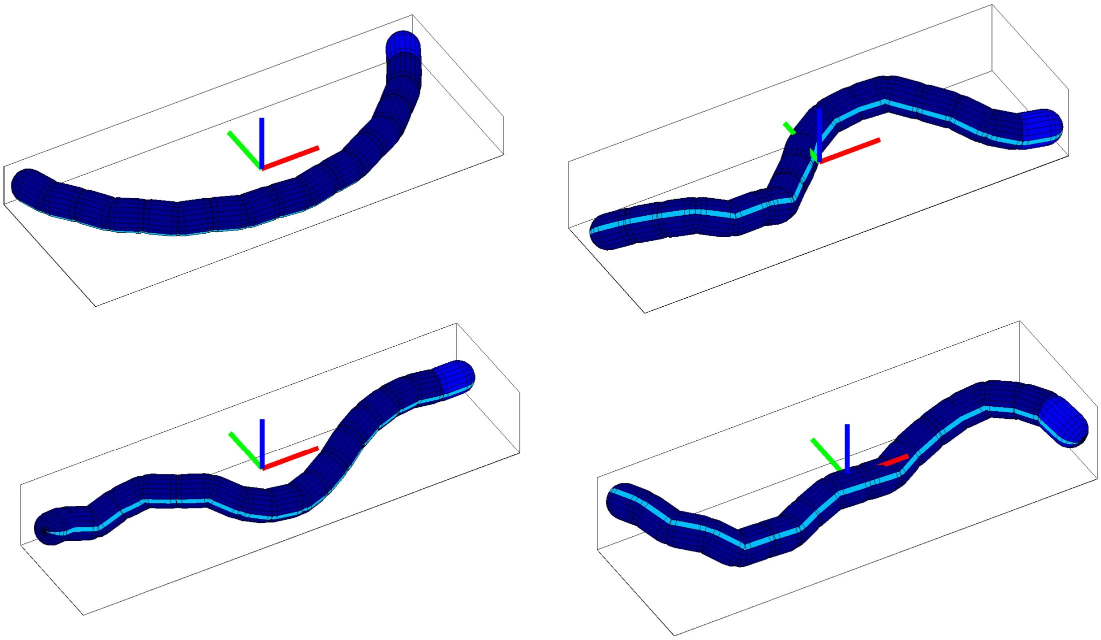

Figure 4.3: An example of the virtual chassis body frame for various shapes of the snake robot. Because the body frame is aligned with the principle components of the robot's shape, it helps separate the robot's internal shape changes from its external motions, improving the performance of lower-fidelity models for state estimation.

fixed to the orientation of the head module.

4.2 State Estimation

Snake robots are unique in both their locomotive capabilities as well as their challenges to estimation and control. Previous work from our group demonstrated accurate estimation of a snake robot's orientation using the robot's proprioceptive sensors [79]. To explore more state of the art techniques, we implemented a UKF and a SSUKF in addition the conventional EKF, although we found that all three methods worked equally well. All three filters used the same process and measurement models, as well as the same values for process and measurement noises. An additional modification from [79] is that this work moves from a second-order to third-order model for the robot's process and measurement models.

4.2.1 Kalman Filter

All of the filters presented in this thesis extend the Kalman filter to non-linear systems. At the heart of the filter are the state estimate, $\hat{\mathbf{x}}_k$, its covariance, $\mathbf{P}_k$, the robot's sensor measurements, $\mathbf{z}_k$, and the non-linear process and measurement models. The process model, f, is a function that predicts the state of the robot given the state at a previous timestep,

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \Delta t). \tag{4.1}$$

The measurement model, h, is a function that generates expected sensor measurements, $\hat{\mathbf{z}}_k$, given the predicted state $\hat{\mathbf{x}}_{k|k-1}$,

$$\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_{k|k-1}). \tag{4.2}$$

In the EKF, the state estimate is propagated through these non-linear functions, and the functions are linearized at each iteration of the filter, resulting in the Jacobians $\mathbf{F}_k$ and $\mathbf{H}_k$ that are first-order approximations of the functions f and h. This allows the propagation of the state covariance between timesteps and the processing of measurements into the state estimate.

In the UKF and SSUKF, a deterministically sampled set of states, called sigma points, are chosen, propagated through the models, and then averaged in a way that directly calculates the state mean and covariance. For an n-dimensional state vector, the UKF uses $2n + 1$ sigma points, and the SSUKF uses $n + 2$ sigma points.

In all of these filters, a process noise matrix, $\mathbf{Q}$, is added to the state covariance at every prediction step. A measurement noise matrix, $\mathbf{R}$, is similarly used to tune the confidence in sensor measurements. The parameters in $\mathbf{Q}$ and $\mathbf{R}$ are used to tune the relative confidence of different states and measurements with respect to each other.

The process model noise for all of the filter variants in this thesis is assumed to be additive, and is incorporated before the state covariance is pushed through the process model at each iteration. In both the EKF [111] and the UKF [51], this allows uncertainty to be added only to the highest derivatives of state variables, relying on the process model to appropriately distribute uncertainty to the lower derivatives.

4.2.2 State Vector

The state of the filter tracks the robot's orientation, its inertial frame acceleration, and its shape variables,

$$\mathbf{x}_k = [\, \mathbf{a}_k \;\; \mathbf{q}_k \;\; \boldsymbol{\omega}_k \;\; \dot{\boldsymbol{\omega}}_k \;\; \boldsymbol{\theta}_k \;\; \dot{\boldsymbol{\theta}}_k \;\; \ddot{\boldsymbol{\theta}}_k \,]^T \tag{4.3}$$

where $\mathbf{a} = [\, a_x \; a_y \; a_z \,]$ is robot's world frame acceleration, $\mathbf{q} = [\, q_1 \; q_2 \; q_3 \; q_4 \,]$ is the quaternion that describes the world frame orientation, $\boldsymbol{\omega} = [\, \omega_x \; \omega_y \; \omega_z \,]$ is the robot's body frame angular velocity, and $\dot{\boldsymbol{\omega}} = [\, \dot{\omega}_x \; \dot{\omega}_y \; \dot{\omega}_z \,]$ is the robot's body frame angular acceleration.

For the rest of the state vector, assume a robot with m links with corresponding joint angles, $\boldsymbol{\theta} = [\, \theta_1, \ldots, \theta_m \,]$, velocities, $\dot{\boldsymbol{\theta}} = [\, \dot{\theta}_1, \ldots, \dot{\theta}_m \,]$, and accelerations, $\ddot{\boldsymbol{\theta}} = [\, \ddot{\theta}_1, \ldots, \ddot{\theta}_m \,]$. This provides a third-order description of how the robot's shape changes over time.

Because the measurement model that generates expected IMU measurements requires the robot's full kinematic state, including the robot's joint angles, velocities, and accelerations in the state vector means that the filter can iterate at every timestep, even if the robot does not report all (or even any) of its joint angles at a given timestep. Furthermore, this formulation of the state means the the joint angles are being redundantly estimated both directly, by observing the joint angles, and indirectly, by observing the inertial readings at adjacent modules. For example, the gravity vectors observed by the accelerometers in two adjacent modules can indicate the joint angle of a module, and the difference in angular velocities observed by the gyros in adjacent modules observe a module's joint angular velocity. In the Section 4.5, we show that this allows us to track the angles of modules that fail to provide feedback, even for extended periods of time.

4.2.3 Process Model

The process model we use for state estimation can be thought of as a generic damped acceleration model that does not explicitly model the interaction of the world. As a third-order model, the highest-order derivative in the state is acceleration.

Acceleration is estimated in a world frame assumed to be damped according to

$$\hat{\mathbf{a}}_k = e^{-\tau \Delta t} \hat{\mathbf{a}}_{k-1}. \tag{4.4}$$

In our experience acceleration must be strongly damped ($\tau \approx 20$) for the filter to perform well on the real robot. We should note that this damping was necessary for estimating the linear acceleration largely because it is extremely difficult to separate out linear acceleration from sensed acceleration acceleration due to gravity using only IMUs. The combination of noisy accelerometers, having only an approximate estimation of orientation, and the fact that gravitational acceleration is typically an order of magnitude greater than linear acceleration means that the estimate of acceleration is usually quite poor. Solutions to this problem usually rely on well-identified models, extremely accurate IMUs, particularly gyros, or observing the robot's linear position or velocity with a visual sensor or motion capture system.

Previous versions of our state estimators [75] actually assumed zero world frame acceleration of the robot. Ignoring body frame acceleration resulted in good performance at slow speeds, but became problematic when the snake robot executed fast motions.

We have found that this damped third-order model provides a tunable compromise between a stable but biased second-order model and the more technically accurate but often unstable third-order model. The estimate of angular acceleration did not need to be damped, since the numerous gyros in the robot directly observe the robot's angular velocity.

The quaternion, $\hat{\mathbf{q}}_k$, representing the orientation of the robot is updated based on the estimated angular velocities at that timestep. We perform a discrete-time update developed by van der Merwe et al. [97]

$$\hat{\mathbf{q}}_k = \exp(-\frac{1}{2}\boldsymbol{\Psi}\Delta t)\ \hat{\mathbf{q}}_{k-1} \tag{4.5}$$

$$\boldsymbol{\Psi} = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \\ -\omega_x & 0 & -\omega_z & \omega_y \\ -\omega_y & \omega_z & 0 & -\omega_x \\ -\omega_z & -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{4.6}$$

The body frame angular velocities of the robot, ω, are assumed to update according to the body frame angular accelerations, and the body frame angular accelerations are assumed to be constant across timesteps,

$$\hat{\boldsymbol{\omega}}_k = \hat{\boldsymbol{\omega}}_{k-1} + \hat{\dot{\boldsymbol{\omega}}}_{k-1}\Delta t \tag{4.7}$$

$$\hat{\dot{\boldsymbol{\omega}}}_k = \hat{\dot{\boldsymbol{\omega}}}_{k-1}. \tag{4.8}$$

The shape of the robot is also estimated in the state of the filter. This is achieved by estimating the snake robot's joint angles, $\hat{\boldsymbol{\theta}}$, their angular velocities, $\hat{\dot{\boldsymbol{\theta}}}$, and their angular accelerations, $\hat{\ddot{\boldsymbol{\theta}}}$. These are updated by according to standard third-order model,

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \hat{\dot{\boldsymbol{\theta}}}_{k-1}\Delta t + \frac{1}{2}\hat{\ddot{\boldsymbol{\theta}}}_{k-1}\Delta t^2. \tag{4.9}$$

$$\hat{\dot{\boldsymbol{\theta}}}_k = \hat{\dot{\boldsymbol{\theta}}}_{k-1} + \hat{\ddot{\boldsymbol{\theta}}}_{k-1}\Delta t. \tag{4.10}$$

$$\hat{\ddot{\boldsymbol{\theta}}}_k = \hat{\ddot{\boldsymbol{\theta}}}_{k-1}. \tag{4.11}$$

The angular velocities of the joints are further modified by a weighted combination of the estimated velocities from the third-order model integration, $\hat{\dot{\boldsymbol{\theta}}}_k$, and the commanded angular velocities at the current timestep, $\dot{\boldsymbol{\theta}}_k^{cmd}$,

$$\hat{\dot{\boldsymbol{\theta}}}_k = (1-\lambda)\hat{\dot{\boldsymbol{\theta}}}_k + \lambda\dot{\boldsymbol{\theta}}_k^{cmd}. \tag{4.12}$$

Even though $\hat{\dot{\boldsymbol{\theta}}}_k$ is modified in this equation we have overloaded the notation, since the filter could be run without this step if commands are unavailable. The weighting parameter λ ranges from 0 to 1 and controls how much of the commanded angular velocity is mixed into the state. A value of 1 essentially overwrites the estimated joint angle velocities at each iteration, whereas a value of 0 turns the filter into a constant-velocity model that has no knowledge of the robot's controls. Since the robot's joints often deviate significantly from their commanded trajectories, this parameter is set relatively low, around 0.25.

4.2.4 Measurement Vector

Our latest snake robot provides feedback measurements from single-axis joint angle encoders, 3-axis accelerometers and a 3-axis gyros located in each module. This means that the vector of measurements in the filter has $7m$ dimensions, where m is the total number of modules in the robot,

$$\mathbf{z}_k = [\ \boldsymbol{\phi}_k\ \ \boldsymbol{\alpha}_k\ \ \boldsymbol{\gamma}_k\ ]^T. \tag{4.13}$$

In (4.13), each element is a vector containing the measurements of a corresponding sensor type for all the modules throughout the snake robot. $\boldsymbol{\phi}$ is the robot's joint angle measurements from its encoders, $\boldsymbol{\alpha}$ is the accelerometer measurements, and $\boldsymbol{\gamma}$ is the gyroscope measurements.

4.2.5 Measurement Model

Our measurement model is a kinematic model that takes into account the robot's shape variables and its inertial state. In the following section, the superscript i indicates the module for which a measurement is predicted and the hat operator denotes a predicted measurement, rather than a sensed measurement from the robot.

Expected joint angle measurements are predicted directly from the estimated angles in the state vector (4.3),

$$\hat{\boldsymbol{\phi}}_k = \hat{\boldsymbol{\theta}}_k. \tag{4.14}$$

Using the estimated joint angles, $\hat{\boldsymbol{\theta}}$, and joint angle velocities, $\hat{\dot{\boldsymbol{\theta}}}$, joint angle accelerations, $\hat{\ddot{\boldsymbol{\theta}}}$, accelerometer and gyro measurements for each module can be predicted using the finite time-differencing approach detailed below. This approach allows us to fuse all of the robot's IMUs into a single estimate of the robot's orientation as well as couple the IMUs to the encoders to redundantly estimate the robot's shape.

Accelerometers have the property that they measure an acceleration due to gravity in addition to lateral acceleration due to motion. For this reason our model treats these two sources of acceleration separately and sums them to generate the predicted accelerometer measurement for each module,

$$\hat{\boldsymbol{\alpha}}_k^i = \hat{\mathbf{a}}_{\text{gravity}}^i + \hat{\mathbf{a}}_{\text{motion}}^i. \tag{4.15}$$

Acceleration due to gravity is predicted by transforming the estimated gravity vector $\hat{\mathbf{g}}$ from the world frame into the frame of each module

$$\hat{\mathbf{a}}_{\text{gravity}}^i = (\hat{\mathbf{W}}_k^i)^T (\hat{\mathbf{V}}_k)^T \hat{\mathbf{g}} \tag{4.16}$$

where $\hat{\mathbf{W}}^i$ is the estimate of the rotation matrix that describes the orientation of module i in the body frame, and $\hat{\mathbf{V}}$ is the estimate of the rotation matrix representation of the quaternion pose $\mathbf{q}$ in the state vector (4.5).

Acceleration due to a module's motion is further split into two components,

$$\hat{\mathbf{a}}_{\text{motion}}^i = \hat{\mathbf{a}}_{\text{internal}}^i + (\hat{\mathbf{W}}_k^i)^T (\hat{\mathbf{V}}_k)^T \hat{\mathbf{a}} \tag{4.17}$$

Acceleration due to the robot's internal shape changes in the body frame, $\hat{\mathbf{a}}_{\text{internal}}^i$, is predicted by double-differentiating the position of the module in the body frame, based on the full kinematic state. Finally, the estimated world frame acceleration of the entire robot is incorporated by rotating the world frame acceleration $\hat{\mathbf{a}}$ from filter's state estimate into the frame of each module.

The predicted gyro measurements for each module are generated by differentiating the orientation of the robot at two nearby timesteps [63]. If $\hat{\mathbf{W}}_k^i$ and $\hat{\mathbf{W}}_{k-1}^i$ are rotation matrices that describe the orientations of module i in the body frame at two timesteps, then gyro measurements due to the robot's motion in the body frame at two timesteps,

k and $k-1$ can be approximated by

$$\begin{bmatrix} 1 & -\bar{\omega}_z^i & \bar{\omega}_y^i \\ \bar{\omega}_z^i & 1 & -\bar{\omega}_x^i \\ -\bar{\omega}_y^i & \bar{\omega}_x^i & 1 \end{bmatrix} \approx \frac{\hat{\mathbf{W}}_k^i (\hat{\mathbf{W}}_{k-1}^i)^T}{\Delta t}. \tag{4.18}$$

The complete prediction for each gyro is the angular velocity from (4.18) plus the robot's body frame angular velocity from the current state estimate, (4.8), rotated into the coordinate from of each module using $\hat{\mathbf{W}}_k^i$

$$\hat{\gamma}_k^i = \bar{\omega}^i + (\hat{\mathbf{W}}_k^i)^T \hat{\omega}_k. \tag{4.19}$$

This finite-differencing approach has allowed us to generate expected readings for all of the robot's IMUs and encoders, based on the current state estimate, $\mathbf{x}$. While our numerical approach somewhat computationally expensive, this part of the measurement model has been implemented in C++ using the Eigen matrix library, allowing the remaining sections of the filter to run in Matlab, in real time, on a standard desktop computer.

4.2.6 State-Based Noise Adjustment

To improve the accuracy of the Gaussian noise assumption for the models and sensors, the additive process and measurement noise matrices, $\mathbf{Q}$ and $\mathbf{R}$, are modified based on the current state estimate. Intuitively, if the robot is moving, the model should be less certain about the current estimate than if the robot is standing still. This matches real-world observation, where it can be readily observed that readings from the robot's sensors, particularly the accelerometers, are much noisier during fast motions than when the robot is still or moving slowly.

The process noise parameters are inflated based on the current state estimate and corresponding tuning parameters. The block of the process noise matrix corresponding to body frame angular acceleration, $\hat{\dot{\boldsymbol{\omega}}}$, is adjusted according to the magnitude of body frame angular acceleration, $\hat{\dot{\boldsymbol{\omega}}}$, and body frame angular velocity $\hat{\boldsymbol{\omega}}$. The additive noise for joint angle accelerations, $\hat{\ddot{\boldsymbol{\theta}}}$, are similarly inflated according to the squared value of the estimated joint angular accelerations, $\hat{\ddot{\boldsymbol{\theta}}}$, and joint angular velocities, $\hat{\dot{\boldsymbol{\theta}}}$.

The measurement noise parameters are also inflated based on the current state estimate. The additive noises for the accelerometers, $\boldsymbol{\alpha}$, in each module are inflated based on the magnitude of the gyro measurements for that module. Of all the modifications in this section, this in particular led to better performance of the filters compared to our previous work, since the accelerometers become particularly noisy when the robot is in motion.

4.2.7 Using Partial Measurement Data

Due to noise in the robot's communications, around 5% of the robot's sensor data is missing at a given update step. In the case of intermittent electrical connections between the modules this percentage can increase even further. Furthermore, when using the snake robots aggressively in the field, modules frequently reset due to their electrical protection circuitry. In these cases an individual module may drop out for 2-3 seconds before rebooting. During this time the module's joint is rotating freely, but there is no direct measurement of its joint angle. Thus, it is desirable to have a method that uses the robot's myriad of other sensors to mitigate these problems.

Missing measurement data can be accommodated in any Kalman filter by only predicting expected measurements for the observed sensors and appropriately resizing the innovation covariance matrix during the filter's update step. An alternative approach that is simple to implement is one where the missing measurement data is replaced with a value of 0, and that measurement's corresponding value in the additive measurement noise matrix, **R**, is increased to 10^6 for that timestep. This causes the filter to effectively ignore the measurement during the update step and is often simpler to implement than dynamically resizing the covariance and state at every iteration.

4.3 Outlier Detection

As mentioned in the previous section, the Kalman filter can be easily modified to accommodate incomplete measurements during its iteration. However, corrupted measurements and outliers that violate the filter's assumption of Gaussian noise are much more problematic [89, 90]. In robotic systems that undergo heavy field use, it is not uncommon for sensors to become unresponsive or miscalibrated. And because these erroneous measurements from an unresponsive sensor can severely disrupt the state estimate, it is beneficial to detect such outliers automatically from the observed sensor data.

4.3.1 Kalman Filter Update

The Kalman filter, and its non-linear variants, updates the estimated state based on a weighted average that takes into account the measurement innovation, which is the difference between the expected measurements, $\hat{\mathbf{z}}_k$, generated from the measurement model, and the observed measurements, $\mathbf{z}_k$, from the robot's sensors,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\hat{\mathbf{z}}_k - \mathbf{z}_k),$$

The proper weighting of the update is determined by the Kalman gain matrix, $\mathbf{K}_k$. In the calculation of this matrix, there is an intermediate step where the innovation covariance, $\mathbf{S}_k$, is calculated. The relevant steps in the respective EKF and UKF /

SSUKF algorithms are presented here for clarity.

In the linear Kalman filter and the EKF, the innovation covariance and Kalman gains are determined based on the estimated covariance and the Jacobian of the measurement model,

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R},$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}.$$

In the UKF and SSUKF, the innovation covariance and Kalman gains are determined directly from matrices, $\mathbf{P}_k^{zz}$ and $\mathbf{P}_k^{xz}$, derived from the sampled sigma points [39], and additive measurement noise,

$$\mathbf{S}_k = \mathbf{P}_k^{zz} + \mathbf{R},$$

$$\mathbf{K}_k = \mathbf{P}_k^{xz} \mathbf{S}_k^{-1}.$$

The Mahalanobis distance for the residual error between the predicted measurement vector, $\hat{\mathbf{z}}_k = \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$, and the observed measurement vector, $\mathbf{z}_k$,

$$d_k = (\hat{\mathbf{z}}_k - \mathbf{z}_k)^T \mathbf{S}_k^{-1} (\hat{\mathbf{z}}_k - \mathbf{z}_k),$$

gives an indication of the likelihood of the measurements.

4.3.2 Algorithm

To detect outliers our method computes, for each sensor s, a Mahalanobis distance that excludes sensor s from the measurement vector. In the following section we will use superscripts $-s$ to clarify that they are quantities where sensor s has been *excluded*. To exclude these elements, we define the following selection matrix,

$$\mathbf{Y}^{-s} = \left[\begin{array}{ccc} \mathbf{I}_{M\times M} & \mathbf{0}_{M\times 3} & \mathbf{0}_{M\times N} \\ \mathbf{0}_{N\times M} & \mathbf{0}_{N\times 3} & \mathbf{I}_{N\times N} \end{array} \right],$$

where M is the number of elements in the measurement vector $\mathbf{z}_k$ that precede sensor $-s$, N is the number of elements in $\mathbf{z}_k$ that follow sensor $-s$, and 3 is the number of elements in the measurement vector that correspond to sensor $-s$ (the $x-y-z$ axes of an accelerometer or gyro).

This process of *excluding* the elements of and calculating the Mahalanobis distance of the remaining elements of $\left[\mathbf{S}_k^{-s}\right]^{-1}$ incorporates information on how the sensors are coupled by the process and measurement models. A simpler, more naive, alternative to this approach would be to calculate the Mahalanobis distance of the 3×3 block of $\left[\mathbf{S}_k^{-s}\right]^{-1}$ corresponding to each sensor directly. This assumes that each sensor is independent, which in some cases may be an appropriate assumption.

The Mahalanobis distance associated with excluding sensor $-s$ can be written as follows,

$$d_k^{-s} = (\hat{\mathbf{z}}_k^{-s} - \mathbf{z}_k^{-s})^T \left[\mathbf{S}_k^{-s}\right]^{-1} (\hat{\mathbf{z}}_k^{-s} - \mathbf{z}_k^{-s}), \tag{4.20}$$

where $\mathbf{z}_k^{-s} = \mathbf{Y}^{-s}\mathbf{z}_k$, $\hat{\mathbf{z}}_k^{-s} = \mathbf{Y}^{-s}\hat{\mathbf{z}}_k$, and $\mathbf{S}_k^{-s} = \mathbf{Y}^{-s}\mathbf{S}_k(\mathbf{Y}^{-s})^T$.

For each excluded sensor $-s$, a Mahalanobis distance d_k^{-s} that is significantly smaller than d_k means that the excluded component is likely to be an outlier. Sorting the set of distances for each sensor ranks the sensor measurements in order of likeliness. We then discard a fixed number of the least likely sensor measurements (the ones that are most likely to be outliers) and calculate the mean, μ_k, and standard deviation, σ_k, of the remaining Mahalanobis distances, presumed to be inliers. For our implementation, we choose to discard 4 sensors from the measurement vector, or one quarter of the robot's sensors.

Finally, for each sensor $-s$, a new metric is computed that compares the Mahalanobis distance d_k^{-s} to the mean and variance of the Mahalanobis distances of the presumed inliers,

$$w_k^{-s} = \frac{\left(d_k^{-s} - \mu_k\right)^2}{\sigma_k^2}.$$

We then consider all sensors, including the initially discarded sensors, and decide whether each one is an inlier or outlier by thresholding the metric w_k^{-s} at some level, ξ. When w_k^{-s} is large, the sensor is likely to be an outlier and when this metric is small, the sensor measurement is likely to be an inlier. If the measurement is determined to be an outlier, the innovation covariance is resized to exclude the measurement, just as is done for missing data. Alternatively, a simple approximation to dynamically resizing the measurement innovation is to simply set the measurement's corresponding element in $\mathbf{R}$ to a comparatively large value, such as 10^6.

In a sense, w_k^{-s} is a Mahalanobis distance of Mahalanobis distances, and is in many ways akin to data clustering. Since this value is based on the overall uncertainty of all of similar sensors in the robot, we are able to pick a single static threshold that is valid at all times. For example, if the process model predicts the state of the robot poorly, or if the filter is poorly tuned, all of the measurements might have a large *absolute* uncertainty. However, this method relies on the *relative* uncertainty between measurements to determine outliers.

Compared to thresholding on the Mahalanobis distance alone, outliers with this method tend to be extremely obvious, often greater than 100 standard deviations from the mean. Setting the detection threshold, ξ, to a value between 10 and 50 has been shown to work well for our system, regardless of sensor type or the robot's motion.

Finally, since this algorithm makes the assumption that some fraction of the sensors must be inliers, it has the benefit that it ensures an upper limit on the number of sensors that can be discarded as outliers, as long as ξ is not set too low (setting $\xi > 3$ standard deviations). This allows outlier detection to be performed even when a filter is still being debugged or is poorly tuned, and the absolute Mahalanobis distances could be unusually large or small.

4.3.3 Efficient Implementation

One drawback of the outlier detection algorithm, as presented thus far, is the need to invert $\mathbf{S}_k^{-s}$ for each sensor in order to compute d_k^{-s} as computed in (4.20). For our 16-link snake robot that has 2 inertial sensors per module, the accelerometer and the gyro, performing this for a typical 16-link robot would require 32 inversions of a 109-by-109 matrix. This is a significant computational expense during real-time operation.

Ideally, it would be beneficial to compute the inverse of the innovation covariance, $\mathbf{S}_k^{-1}$, only once and then to somehow efficiently infer, for each sensor s, the matrix $\left[\mathbf{S}_k^{-s}\right]^{-1}$. By definition, $\left[\mathbf{S}_k^{-s}\right]^{-1} = \left[\mathbf{Y}^{-s}\mathbf{S}_k(\mathbf{Y}^{-s})^T\right]^{-1}$. But unfortunately, $\left[\mathbf{S}_k^{-s}\right]^{-1} \neq \mathbf{Y}^{-s}\mathbf{S}_k^{-1}(\mathbf{Y}^{-s})^T$, otherwise we would compute $\left[\mathbf{S}_k^{-s}\right]^{-1}$ directly from $\mathbf{S}_k^{-1}$. This is not possible because $\mathbf{Y}^{-s}$ is not an orthogonal matrix.

Instead, we can leverage the Woodbury matrix identity [72], to perform a low-rank correction to the relatively large matrix $\mathbf{S}_k^{-1}$. This allows us to efficiently obtain the matrix $\left[\mathbf{S}_k^{-s}\right]^{-1}$, which we require for computing the Mahalanobis distance in Eq. (4.20). First, we define the following matrix that can be used to rearrange the elements of the innovation so that the elements corresponding with sensor $-s$ are last,

$$\mathbf{G}^{-s} = \begin{bmatrix} \longleftarrow & \mathbf{Y}^{-s} & \longrightarrow \\ \mathbf{0}_{3xM} & \mathbf{I}_{3x3} & \mathbf{0}_{3xN} \end{bmatrix},$$

Using this matrix, we can rearrange the inverse of the innovation covariance matrix as follows,

$$[\mathbf{S}'_k]^{-1} = \left[\mathbf{G}^{-s}\mathbf{S}_k(\mathbf{G}^{-s})^T\right]^{-1} = \mathbf{G}^{-s}\mathbf{S}_k^{-1}(\mathbf{G}^{-s})^T = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}.$$

Using the Woodbury matrix identity to invert the matrix $[\mathbf{S}'_k]^{-1}$, we obtain,

$$\mathbf{S}'_k = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)^{-1} & \sim \\ \sim & \sim \end{bmatrix}.$$

Above, the $\sim$ indicates regions of the matrix from the Woodbury matrix identity that are omitted for clarity. Since the upper left component of $\mathbf{S}'_k$ is equal to the matrix $\mathbf{S}_k^{-s}$, due the existence of $\mathbf{Y}^{-s}$ in $\mathbf{G}^{-s}$, we can simply invert the upper left component of $\mathbf{S}'_k$,

$$\left[\mathbf{S}_k^{-s}\right]^{-1} = \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T. \tag{4.21}$$

In (4.21), $\mathbf{C}$ is the inverted covariance of the sensor being excluded. This form only requires the inversion of $\mathbf{C}$, which in our case is 3-by-3. Thus, we can efficiently calculate the inverse of the innovation covariance, $\left[\mathbf{S}_k^{-s}\right]^{-1}$ for each test of a sensor s by performing a small update to the full inverse of the innovation covariance matrix $\mathbf{S}_k^{-1}$. A summary of the algorithm that we use for outlier detection is provided in Algorithm 1.

4.4 Experiment

To test the accuracy of the different state estimators, multiple trials of the snake robot were performed in a Vicon motion capture system. During these trials, the robot was remotely controlled through a wide variety of motions, some of which were quite fast

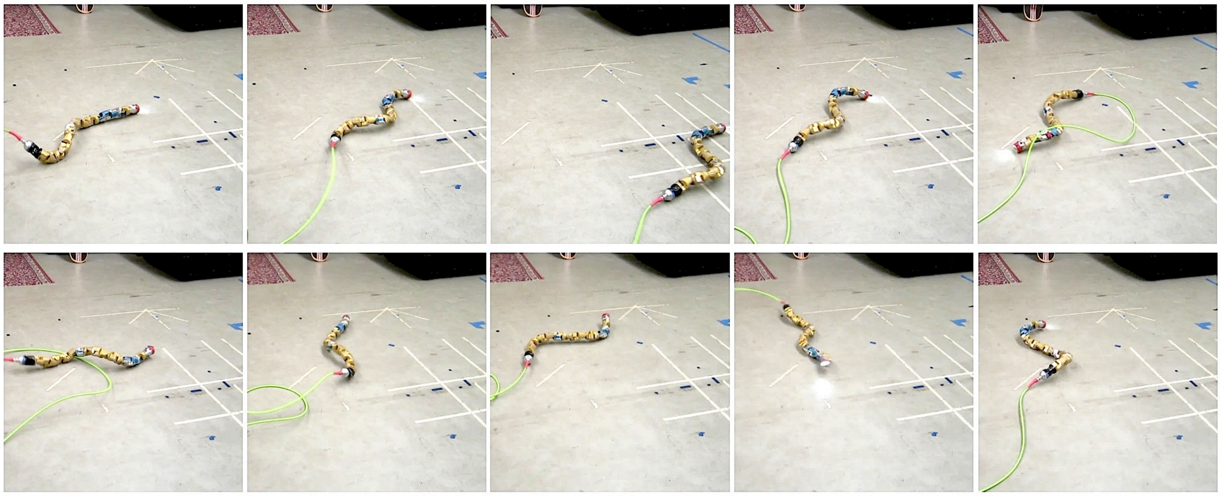

Figure 4.4: A montage of the snake robot's movements in one of the motion capture trials. The robot does a combination of motions that include slithering forward, sidewinding right and left, and turning in place clockwise and counter-clockwise. This montage corresponds to the plots that are presented in the results section.

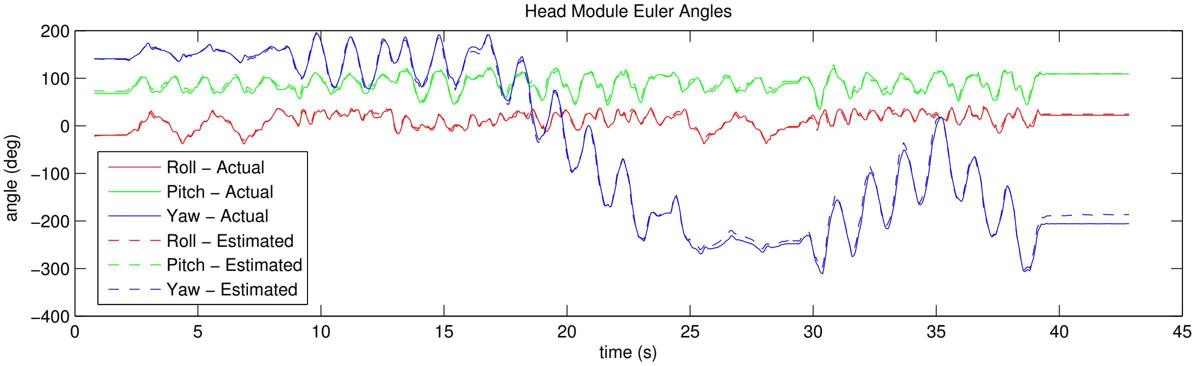

Figure 4.5: A comparison of the orientation of the head module from motion capture (solid line) compared to the state estimate of the filter (dashed line). The results presented are for the SSUKF, although the UKF and EKF performed similarly.

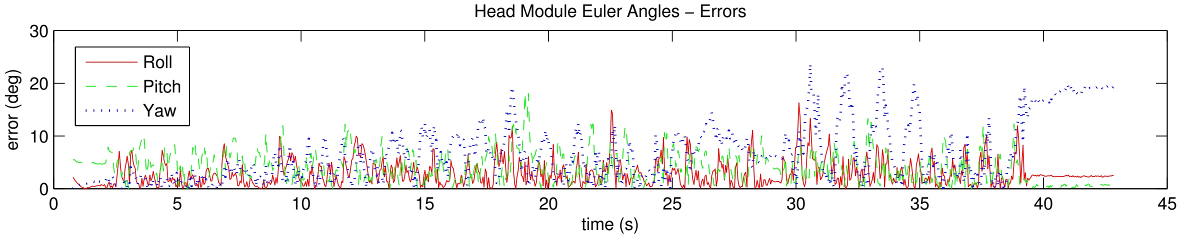

Figure 4.6: The error of the state estimate compared to the motion capture data for the SSUKF. Note that the error in yaw increases over time because it is being dead-reckoned from gyros.

Algorithm 1 Outlier Detection Procedure

```
for each $-s$ do
    $\mathbf{S}_k'^{-1} \leftarrow \mathbf{G}^{-s}\mathbf{S}_k^{-1}(\mathbf{G}^{-s})^T$
    $[\mathbf{A}, \mathbf{B}, \mathbf{C}] \leftarrow \text{extractBlocks}\left(\mathbf{S}_k'^{-1}\right)$
    $\left[\mathbf{S}_k^{-s}\right]^{-1} \leftarrow \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$
    $\mathbf{z}_k^{-s} = \mathbf{Y}^{-s}\mathbf{z}_k$
    $\hat{\mathbf{z}}_k^{-s} = \mathbf{Y}^{-s}\hat{\mathbf{z}}_k$
    $d_k^{-s} = (\hat{\mathbf{z}}_k^{-s} - \mathbf{z}_k^{-s})^T \left[\mathbf{S}_k^{-s}\right]^{-1} (\hat{\mathbf{z}}_k^{-s} - \mathbf{z}_k^{-s})$
end for
$[\mu_k, \sigma_k] \leftarrow$ statistics_of_inliers$\left(d_k^{-s}\right.$ for all $\left.s\right)$
for each $-s$ do
    $w_k^{-s} \leftarrow \frac{\left(d_k^{-s} - \mu_k\right)^2}{\sigma_k^2}$
    if $w_k^{-s} > \xi$ then
        Mark $s$ as outlier
    end if
end for
```

and abrupt. The pose of the head module was tracked by the motion capture system to provide ground truth. This was then compared to the filter's estimate for pose of the head module, which is dependent on the estimate of the entire state of the robot. Figure 4.5 shows a montage of the robot during one of these trials.

To demonstrate the advantages of redundant state formulation, we simulated missing data and complete module dropouts. To test the outlier detection algorithm, we simulated corrupted data on the robot's inertial sensors similar to what is seen when a module is poorly calibrated or programmed incorrectly.

4.5 Results

Overall, the EKF, UKF and SSUKF performed comparably. The filters were all able to run in real time on the feedback data coming from the robot, about 20 Hz. A comparison of the errors of the estimated head module orientation, converted to Euler angles, for each filter is presented in Table 4.1. This is the averaged error for 3 different

Filter Performance Comparison			
Euler Angle Errors (degrees)			
	Roll	Pitch	Yaw
EKF	3.3	3.9	13.4
UKF	3.2	3.8	11.3
SSUKF	3.2	3.8	10.9

Table 4.1: The accuracy of various non-linear Kalman filters in estimating the Euler angle orientation of the head module of the snake robot. All of the filters perform comparably.

motion capture trials where the robot was driven in a wide variety of speeds and directions. The accuracy in yaw is significantly worse than pitch and roll because it is being dead-reckoned based on the filter's integration of estimated angular velocities.

Figures 4.5 and 4.6 show a comparison of the estimated head module orientation from the SSUKF compared the motion capture data for one of the trials. To provide a meaningful comparison the quaternion orientations of the head have been converted into Euler angles.

4.5.1 State-Based Noise Adjustment

Table 4.2 shows the performance of the EKF, UKF, and SSUKF without dynamically adjusting the process and measurement noises based on the estimated state, as described in Section 4.2.6. There was no measurable benefit of tuning the noises dynamically for pitch and roll, and all of the filters performed comparably. However, there was a significant improvement in the dead-reckoned yaw performance for all of the filters. We believe this is primarily because the state-based noise adjustment allows the gyros being trusted more heavily for fast motions, while at the same time trusting the more noisy accelerometers less.

Filter Performance - Without Dynamic Noise Adjustment

Euler Angle Errors (degrees)

	Roll	Pitch	Yaw
EKF	2.9	3.4	36.3
UKF	2.9	3.4	34.7
SSUKF	3.1	3.3	24.3

Table 4.2: The accuracy of the various filters in estimating the orientation of the head module of the snake robot, without dynamically adjusted process and measurement noise. The filters perform similarly as when noises are not dynamically tuned, with the exception that dead-reckoned yaw estimation is significantly worse.

SSUKF Performance - Missing Data

Euler Angle Errors (degrees)

	Roll	Pitch	Yaw
Baseline	3.2	3.8	10.9
25% Missing	3.6	3.9	9.4
50% Missing	5.6	5.8	26.5
75% Missing	9.0	11.1	57.5

Table 4.3: The accuracy of the SSUKF in predicting the Euler angle orientation of the head module of the snake robot. The filter performs well even with half of the robot's data being excluded. Even when 50% of the data is excluded the filter continues to run well, although accuracy begins to be degraded for larger amounts of feedback loss.

4.5.2 Partial Measurement Data

Under normal circumstances, our snake robot drops about 5% of its data due noise and errors in its communications. To simulate more adverse conditions, we randomly selected and removed higher proportions of the robot's feedback data. The results are summarized in Table 4.3.

We simulated prolonged module dropouts by eliminating all of the feedback data (joint angles, gyros, accelerometers) from a module in the robot for the entirety of the same data set shown in Fig. 4.5 and Fig. 4.6. For the data presented here, the joint angles and inertial sensors were unavailable for the entire run in modules 3, 6, 7 and 12. Even if up to 4 modules were eliminated the filter still converged and estimated

SSUKF Performance - Dropped Modules			
Euler Angle Errors (degrees)			
	Roll	Pitch	Yaw
Baseline	3.2	3.8	10.9
Missing 4 Modules	4.0	4.3	59.1

Table 4.4: The accuracy of the SSUKF in the presence of missing data from multiple modules for the entire run. These results are for the same trial as shown in Fig. 4.4. Feedback from a quarter of the snake robot (modules 3,6,7 and 12) was eliminated.

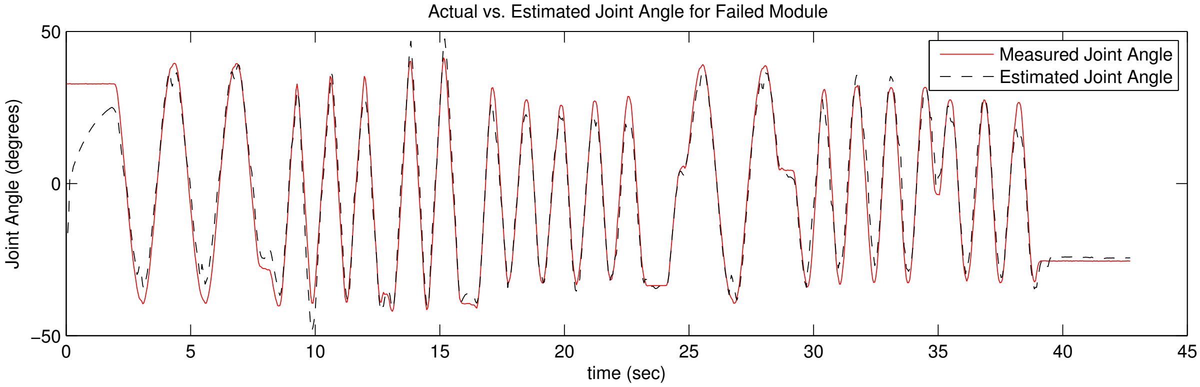

Figure 4.7: A comparison of the actual and estimated joint angle for module 7 in the snake robot. No feedback was available to the filter for modules 3,6,7 and 12 during the entire trial, but the joint angle is able to be estimated from the feedback from the remaining modules.

the pose of the head reasonably well (Table 4.4).

The accuracy of a missing joint angle being estimated by the filter is shown in more detail by Figs. 4.7 and 4.8. While the error of the estimated joint angle is as much as 10° at times, the filter does a reasonable job of estimating the joint angle even while the robot is undergoing significant motion. It is also worth noting that the estimated uncertainty of the joint angle is appropriately captured by the filter.

4.5.3 Different Body Frames

Based on our previous experience, we believe that the choice of body frame can effect the accuracy of state estimation, and as such we ran the SSUKF in four different body frames. As a baseline comparison, the first body frame was fixed to the head module

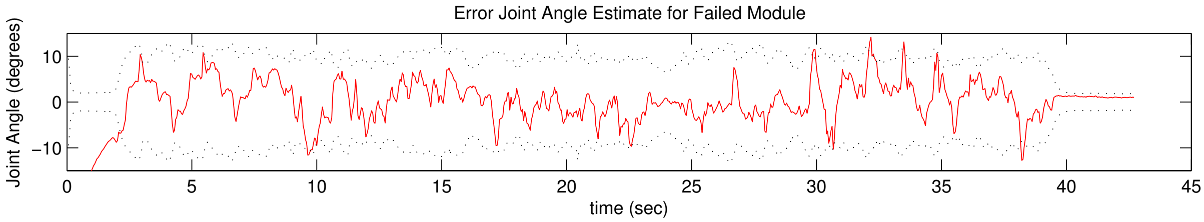

Figure 4.8: The error for estimating a missing joint angle for an entire trial. The red line is the error of the estimated angle, and the dotted lines are the 3σ bounds of the estimated covariance. After the filter converges it tracks the joint angle reasonably well, with a mean absolute error of 3 degrees.

of the robot. The second was a body frame where the orientation was fixed to the head module, while the origin was located at the geometric center of the robot. The last two body frames were the original virtual chassis [77] and the modified virtual chassis presented in Appendix A.

The results of using these different body frames in the SSUKF are presented in Table 4.6. Compared to our previous work that used simpler process models [75], the benefits of an averaged body frame over a fixed frame were less pronounced. However, we do note that the three averaged body frames provide a varying degrees of improvement in dead-reckoning the robot's yaw. We suspect that this improvement is due to the naive third-order kinematic model being a more accurate prediction of the averaged motion of the robot than the motion of any individual module.

4.5.4 Redundant IMUs

Finally, we wanted to better quantify any improvements in estimating the robot's orientation that result from having a large number of IMUs distributed throughout the robot. Figure 4.9 shows the error of the estimated head module orientation using increasing numbers of IMUs, starting at the first module behind the head and adding more IMUs tail-ward.

It is interesting to note that there is very little improvement in the estimated pitch

SSUKF Performance - Corrupted IMU Data **Euler Angle Errors (degrees)**			
	Roll	Pitch	Yaw
Outlier Detection OFF	6.7	4.7	30.7
Outlier Detection ON	3.3	3.8	12.4
Baseline (Clean Data)	3.2	3.8	10.9

Table 4.5: The accuracy of the SSUKF when accelerometer and gyro feedback from quarter of the robot (modules 3, 6, 7, and 12) was corrupted. With the outlier detection, the filter performs almost as well as with clean data.

and roll, while there is a dramatic improvement in dead-reckoned yaw. This indicates that for practical purposes the main limitation of low-cost IMUs is the bias drift of the gyros. The improvement in yaw by incorporating n gyros roughly follows the $\sqrt{n}$ trend we would expect, up until about 8 modules. We suspect that past this point, the noise in the velocities of the robot's joint angles limits the usefulness of having the gyros distributed throughout the moving modules of the robot. However, we should point out that having these extra IMUs is still advantageous for making up for missing joint angle measurements or failed modules as shown in Section 4.2.7.

4.5.5 Outlier Detection

To test our method of outlier detection, feedback data from the IMUs was corrupted by having its sign reversed for the entire trial. For the data presented in Table 4.5, modules 3, 6, 7 and 12 had their IMU data corrupted. When running the outlier detection, the filter performs on par with its normal baseline performance. However, it is worth noting that the redundant state formulation is robust enough to remain stable even without the outlier detection, albeit with degraded performance.

SSUKF Performance - Different Body Frames Euler Angle Errors (degrees)			
	Roll	Pitch	Yaw
Head-Fixed Frame	5.4	4.2	17.4
Head-Fixed Orientation / COM	4.1	4.1	14.0
Virtual Chassis [76]	3.2	3.8	10.0
Alternate Virtual Chassis	3.1	3.6	10.8

Table 4.6: The accuracy of various of the SSUKF in estimating the Euler angle orientation of the head module of the snake robot using different body frames for the filter process and measurement models. An averaged body frame improves performance slightly, although the amount improvement depends heavily on the nature of the robot's motion.

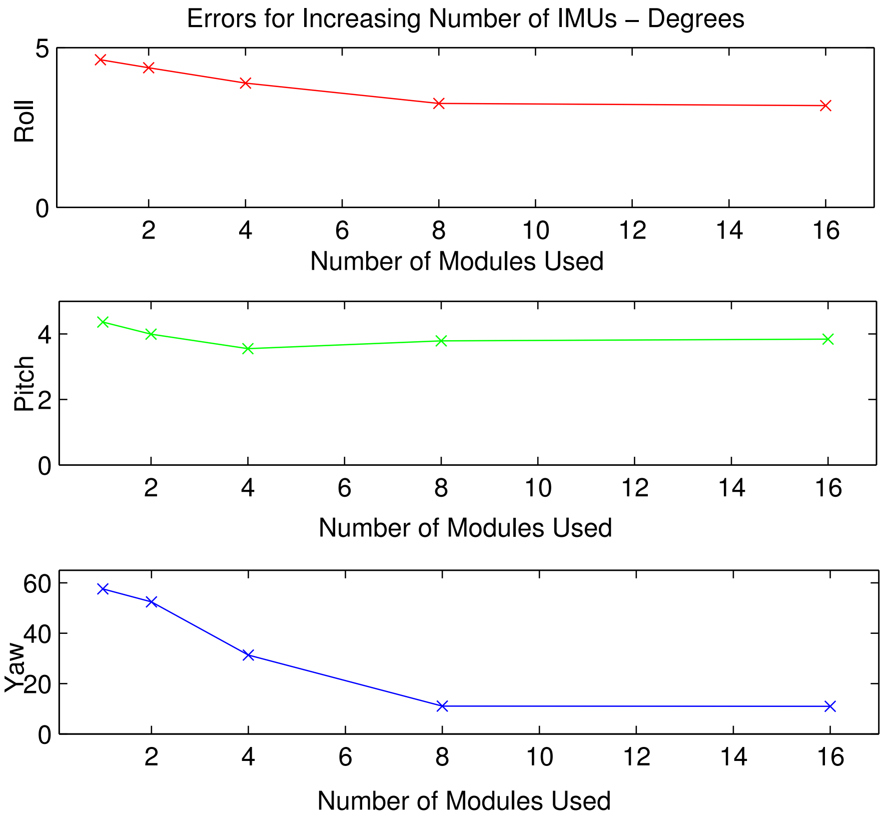

Figure 4.9: Error of the estimated head module orientation using increasing numbers of IMUs for the state estimate. Using additional IMUs provides little improvement in the estimation of pitch and roll, but greatly reduces the dead-reckoning error in yaw.

Bibliography

[1] P. Abbeel, A. Coates, M. Montemerlo, A. Ng, and S. Thrun. Discriminative training of Kalman filters. In *Robotics: Science and Systems*, 2005. 2.4.1

[2] A. M. Andruska and K. S. Peterson. Control of a Snake-Like Robot in an Elastically Deformable Channel. *IEEE/ASME Transactions on Mechatronics*, 13(2):219–227, Apr. 2008. 6.2

[3] J. Badger, A. Hulse, R. Taylor, A. Curtis, D. Gooding, and A. Thackston. Model-based Robotic Dynamic Motion Control for the Robonaut 2 Humanoid Robot. In *IEEE/RAS International Conference on Humanoid Robotics*, volume 2, Atlanta, USA, 2013. 11.3

[4] D. C. Bentivegna and C. G. Atkeson. Compliant control of a hydraulic humanoid joint. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 483–489. Ieee, Nov. 2007. 2.4.1

[5] J. Borenstein, M. Hansen, and A. Borrell. The OmniTread OT-4 Serpentine RobotâĂŤDesign and Performance. *Journal of Field Robotics*, 24(7):601–621, 2007. 6.2

[6] R. Bro, E. Acar, and T. Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics*, 22(2):135–140, 2008. A.2.1

[7] J. Burdick, J. Radford, and G. Chirikjian. A Sidewinding Locomotion Gait for Hyper-Redundant Robots. *Robotics and Automation*, 3:101–106, 1993. 2.4.1, A.3.1

[8] J. G. Castrejon Lozano, L. R. Garca Carrillo, A. Dzul, and R. Lozano. Spherical simplex sigma-point Kalman filters: A comparison in the inertial navigation of a terrestrial vehicle. *2008 American Control Conference*, pages 3536–3541, June 2008. 2.4.2

[9] G. Chirikjian and J. Burdick. A modal approach to hyper-redundant manipulator kinematics. *IEEE Transactions on Robotics and Automation*, 10(3):343–354, June 1994. 2.1, 2.4.1

[10] G. Chirikjian and J. Burdick. The kinematics of hyper-redundant robot locomotion. *IEEE Transactions on Robotics and Automation*, 11(6):781–793, 1995. 2.1

[11] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory Algorithms, and Implementations*. MIT Press, Boston, USA, 2005. 2.3.1, 2.4.2

[12] H. Date and Y. Takita. Adaptive locomotion of a snake like robot based on curvature derivatives. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3559, San Diego, USA, Oct. 2007. IEEE. 6.3, 9.2.1, 11.2

[13] E. Delponte, F. Isgrò, F. Odone, and A. Verri. SVD-matching using SIFT features. *Graphical Models*, 68(5-6):415–431, Sept. 2006. A.1

[14] M. Diftler, J. Mehling, M. Abdallah, N. Radford, L. Bridgwater, A. Sanders, R. Askew, D. Linn, J. Yamokoski, F. Permenter, B. Hargrave, R. Platt, R. Savely, and R. Ambrose. Robonaut 2-the first humanoid robot in space. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 2178–2183, Shanghai, China, 2011. 6.1, 6.1, 6.2

[15] Z. M. Durovic and B. D. Kovacevic. Robust estimation with unknown noise statistics. *Automatic Control, IEEE Transactions on*, 44(6):1292–1296, June 1999. 2.4.2

[16] F. Enner, D. Rollinson, and H. Choset. Simplified Motion Modeling for Snake Robots. In *IEEE International Conference on Robotics and Automation*, St. Paul, USA, 2012. 4.1, 11.2, A.6.3

[17] F. Enner, D. Rollinson, and H. Choset. Motion Estimation of Snake Robots in Straight Pipes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5148–5153, Karlsruhe, Germany, 2013. 4.1, 11.2, A.3.5

[18] A. Fod, M. Matarić, and O. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, 2002. 11.4

[19] S. Ford, D. Rollinson, A. Willig, and H. Choset. Online Calibration of a Compact Series Elastic Actuator. In *American Controls Conference (ACC)*. AACC, 2014. 11.3, B.1.4

[20] A. N. Gent. Engineering with Rubber: How to Design Rubber Components. In *Engineering with Rubber: How to Design Rubber Components*, chapter 3. Hanser Publications, 2nd edition, 2001. 7.1.2

[21] D. I. Goldman and D. L. Hu. Wiggling Through the World: The mechanics of slithering locomotion depend on the surroundings. *American Scientist*, pages 314–323, 2010. 6.3

[22] J. Gonzalez-Gomez, H. Zhang, E. Boemo, and J. Zhang. Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules. In *9th international conference on climbing and walking robots*. Citeseer, 2006. 2.1

[23] J. Gray. The Mechanism of Locomotion in Snakes. *Journal of Experimental Biology*, 23(2):101–123, Dec. 1946. 6.2

[24] E. Guizzo and E. Ackerman. The Rise of the Robot Worker. *IEEE Spectrump*, (october):34–41, Oct. 2012. 6.1

[25] Z. V. Guo and L. Mahadevan. Limbless undulatory propulsion on land. *Proceedings of the National Academy of Sciences of the United States of America*, 105(9):3179–

84, Mar. 2008. 6.2, 6.3

[26] M. Hara, S. Satomura, H. Fukushima, T. Kamegawa, H. Igarashi, and F. Matsuno. Control of a Snake-like Robot Using the Screw Drive Mechanism. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 28(3):541–554, Apr. 2007. 2.1

[27] R. L. Hatton and H. Choset. Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction. *Autonomous Robots*, 28(3):271–281, Dec. 2009. 9.1

[28] R. L. Hatton and H. Choset. Geometric motion planning: The local connection, Stokes's theorem, and the importance of coordinate choice. *The International Journal of Robotics Research*, June 2011. A.1

[29] B. Y. S. Hirose and H. Yamada. Snake-Like Robots. *IEEE Robotics & Automation Magazine*, (March):88–98, 2009. 2.1

[30] S. Hirose. *Biologically Inspired Robots*. Oxford University Press, 1993. 2.1, 2.2, 2.4.1, 6.2, 6.3, 9.2

[31] D. L. Hu, J. Nirody, T. Scott, and M. J. Shelley. The mechanics of slithering locomotion. *Proceedings of the National Academy of Sciences of the United States of America*, 106(25):10081–5, June 2009. 6.3

[32] J. Hurst, D. Hobbelen, and A. Rizzi. Series Elastic Actuation: Potential and Pitfalls. In *International Conference on Climbing and Walking Robots (CLAWAR).*, 2004. 2.4.1

[33] J. W. Hurst. *The Role and Implementation of Compliance in Legged Locomotion*. PhD thesis, Carnegie Mellon University, 2008. 6.1, 8.1, 8.2

[34] A. Ijspeert and A. Crespi. Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 262–268. Ieee, Apr. 2007. 2.1, 2.4.1

[35] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–53, May 2008. 2.4.1

[36] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. *Advances in Neural Information Processing Systems*, 15:1523–1530, 2002. 11.4

[37] B. Jayne. Kinematics of Terrestrial Snake Locomotion. *Copeia*, pages 915–927, 1986. 6.3

[38] B. Jones and I. Walker. Kinematics for multisection continuum robots. *Robotics, IEEE Transactions on*, 22(1):43–55, 2006. 6.2

[39] S. Julier. The spherical simplex unscented transformation. *Proceedings of the 2003 American Control Conference, 2003.*, pages 2430–2434, 2003. 2.4.2, 4.3.1

[40] S. Julier and J. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceed-*

ings of the IEEE, 92(3):401–422, Mar. 2004. 2.4.2

[41] S. W. Jung, K. M. Ryu, S. G. Choi, and H. R. Roh. Inpipe Inspection Robot System with Active Steering Mechanism. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1652–1657, 2000. 2.4.3

[42] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1644–1650. IEEE, 2003. A.1, A.6.1

[43] S. Kajita and K. Tan. Study of Dynamic Biped Locomotion on Rugged Terrain - Derivation and Application of the Linear Inverted Pendulum Mode -. In *IEEE International Conference on Robotics and Automation (ICRA)*, number April, pages 1405–1411, Sacramento, USA, 1991. IEEE. 2.4.1

[44] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 2.4.2

[45] T. Kamegawa, T. Harada, and A. Gofuku. Realization of cylinder climbing locomotion with helical form by a snake robot with passive wheels. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3067–3072, Kobe, Japan, 2009. 2.1

[46] T. Kamegawa, R. Kuroki, M. Travers, and H. Choset. Proposal of EARLI for the snake robot's obstacle aided locomotion. *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Nov. 2012. 6.3

[47] T. Kano and A. Ishiguro. Obstacles are beneficial to me! Scaffold-based locomotion of a snake-like robot using decentralized control. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3273–3278, Tokyo, Japan, Nov. 2013. IEEE. 6.3, 11.2

[48] F. Karlsson and A. Persson. *Modelling Non-Linear Dynamics of Rubber Bushings - Parameter Identification and Validation*. PhD thesis, Lund University, 2003. 7.2

[49] M. Kikuchi and I. A. N. D. Aiken. An Analytical Hysteresis Model for Elastomeric Seismic Isolation Bearings. *Earthquake Engineering and Structural Dynamics*, 26:215–231, 1997. 11.3

[50] K. Kong, J. Bae, and M. Tomizuka. A compact rotary series elastic actuator for knee joint assistive system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2940–2945, Anchorage, USA, 2010. 6.1

[51] E. Kraft. A quaternion-based unscented Kalman filter for orientation tracking. *Proceedings of the Sixth International Conference on Information Fusion*, 1:47–54, 2003. 4.2.1

[52] A. D. Kuo. The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor control*, 6(2):129–45, Apr. 2002. 2.4.1

[53] A. Kuwada, S. Wakimoto, K. Suzumori, and Y. Adomi. Automatic pipe negotia-

tion control for snake-like robot. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, (438):558–563, July 2008. 2.4.3

[54] W. Lee. *Designing articulated legs for running machines*. PhD thesis, Massachusetts Institute of Technology, 1990. 7.1, 7.1

[55] M. Li, B. H. Kim, and A. I. Mourikis. Real-time Motion Tracking on a Cellphone using Inertial Sensing and a Rolling-Shutter Camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4697–4704, Karlsruhe, Germany, 2013. 11.2

[56] M. Li and A. I. Mourikis. 3-D Motion Estimation and Online Temporal Calibration for Camera-IMU Systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5689–5696, Karlsruhe, Germany, 2013. 11.2

[57] P. Liljeback, K. Pettersen, O. Stavdahl, and J. Gravdahl. Experimental Investigation of Obstacle-Aided Locomotion With a Snake Robot. *Robotics, IEEE Transactions on Robotics*, 27(4):792–800, 2011. 2.4.1, 6.4, 6.3, 11.2

[58] R. Luo, C. Yih, and K. Su. Multisensor fusion and integration: approaches, applications, and future research directions. *Sensors Journal, IEEE*, 2(2):107–119, 2002. 2.4.2

[59] F. L. Markley. Attidude Determination using Vector Observations and the Singular Value Decomposition. *The Journal of Astronautical Sciences*, 36(3):245–258, 1988. A.5

[60] M. T. Mason. *Compliance and Force Control for Computer Controlled Manipulators*. PhD thesis, Massachusetts Institute of Technology, 1979. 2.4.1

[61] W. Mosauer. A Note on the Sidewinding Locomotion of Snakes. *The American Naturalist*, 64(691):179–183, Mar. 1930. A.3.1

[62] W. Mosauer. On the Locomotion of Snakes. *Science*, 76(1982):583–585, 1932. 6.2, 6.3

[63] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994. 2.3.1, 4.2.5, A.2.1

[64] A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7:308–313, 1965. A.3.3

[65] M. Nordin and P. Gutman. Controlling mechanical systems with backlashâĂŤa survey. *Automatica*, 38:1633–1649, 2002. 11.3

[66] H. Ohno and S. Hirose. Design of slim slime robot and its gait of locomotion. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pages 707–715, 2001. 2.1, 2.2, 6.2, 9.3, 9.3.2

[67] Y. Peng, D. Vrancic, and R. Hanus. Anti-Windup, Bumpless, and Conditioned Transfer Techniques for PID Controllers. *IEEE Control Systems*, pages 48–57, 1996. B.3.3

[68] L. Pfeffer, O. Khatib, and J. Hake. Joint Torque Sensory Feedback in the Control

of a PUMA Manipulator. *IEEE Transactions on Robotics and Automation*, 5(4):418–425, 1989. 2.4.1

[69] M. Pilu. A direct method for stereo correspondence based on singular value decomposition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, number 224, pages 261–266. IEEE Comput. Soc, 1997. A.1

[70] G. Pratt and M. Williamson. Series elastic actuators. *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pages 399–406, 1995. 2.4.1, 6.1

[71] G. Pratt and P. Willisson. Late motor processing in low-impedance robots: Impedance control of series-elastic actuators. In *American Controls Conference (ACC)*, pages 3245 – 3251, Boston, USA, 2004. 11.3

[72] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992. 4.3.3

[73] D. W. Robinson. *Design and Analysis of Series Elasticity in Closed-loop Actuator Force Control by*. PhD thesis, Massachusetts Institute of Technology, 2000. 8.1, 8.2

[74] D. W. Robinson, J. E. Pratt, D. J. Paluska, and G. A. Pratt. Series elastic actuator development for a biomimetic walking robot. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, USA, 1999. 2.4.1, 6.1

[75] D. Rollinson, A. Buchan, and H. Choset. State Estimation for Snake Robots. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1075—-1080, San Francisco, USA, 2011. 2.4.1, 4.1, 4.2.3, 4.5.3, 10

[76] D. Rollinson, A. Buchan, and H. Choset. Virtual Chassis for Snake Robots: Definition and Applications. *Advanced Robotics*, pages 1–22, Oct. 2012. 4, 4.1, **??**

[77] D. Rollinson and H. Choset. Virtual Chassis for Snake Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, USA, 2011. 4.5.3, 11.4

[78] D. Rollinson and H. Choset. Gait-Based Compliant Control for Snake Robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5123–5128, Karlsruhe, Germany, 2013. 5.1, 9.3

[79] D. Rollinson, H. Choset, and S. Tully. Robust State Estimation with Redundant Proprioceptive Sensors. In *ASME Dynamic Systems and Control Conference (DSCC)*, Palo Alto, USA, 2013. 4, 4.2, 5.1, 11.2, 11.4

[80] D. Rollinson, S. Ford, B. Brown, and H. Choset. Design and Modeling of a Series Elastic Element for Snake Robots. In *ASME Dynamic Systems and Control Conference (DSCC)*, Palo Alto, USA, 2013. 8.1, B.1.4

[81] I. Schick and S. Mitter. Robust Recursive Estimation in the Presence of Heavy-Tailed Observation Noise. *The Annals of Statistics*, 22(2):1045–1080, 1994. 2.4.2

[82] G. L. Scott and H. C. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings. Biological sciences / The Royal Society*, 244(1309):21–6,

Apr. 1991. A.1

[83] N. Seegmiller, F. Rogers-Marcovitz, and A. Kelly. Online calibration of vehicle powertrain and pose estimation parameters using integrated dynamics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3969–3974. Ieee, May 2012. 2.4.1

[84] A. Seyfarth, H. Geyer, R. Blickhan, and S. Lipfert. Running and walking with compliant legs. *Lecture Notes in Control and Information Sciences: Fast Motions in Biomechanics and Robotics*, 340:383–401, 2006. 6.1

[85] A. Shapere and F. Wilczek. Geometry of self-propulsion at low Reynolds number. *Journal of Fluid Mechanics*, 198:557–585, Apr. 1989. A.1

[86] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer, London, 2009. 8.1

[87] S. Takaoka, H. Yamada, and S. Hirose. Snake-like active wheel robot ACM-R4.1 with joint torque sensor and limiter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1081–1086, San Francisco, USA, Sept. 2011. Ieee. 6.2

[88] M. Tesch, K. Lipkin, I. Brown, R. L. Hatton, A. Peck, J. Rembisz, and H. Choset. Parameterized and Scripted Gaits for Modular Snake Robots. *Advanced Robotics*, 23(9):1131–1158, June 2009. 2.1, 2.2, 2.4.1, 5.1, 5.1, 6.3, 9.3, 9.3.2

[89] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005. 2.4.2, 4.3

[90] J.-a. Ting, E. Theodorou, and S. Schaal. Learning an Outlier-Robust Kalman Filter. In *Machine Learning: ECML*, pages 748–756., Warsaw, Poland, 2007. University of Southern California. 2.4.2, 4.3

[91] A. A. Transeth. *Modelling and Control of Snake Robots*. PhD thesis, Norwegian University of Science and Technology (NTNU), 2007. 2.1, 6.3

[92] A. A. Transeth, K. Y. Pettersen, and P. l. Liljebäck. A survey on snake robot modeling and locomotion. *Robotica*, 27(07):999, Mar. 2009. 2.1

[93] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. 1997. A.2.1

[94] J. Urata, T. Hirose, and Y. Namiki. Thermal control of electrical motors for high-power humanoid robots, Sept. 2008. B.3.4

[95] J. Urata, Y. Nakanishi, K. Okada, and M. Inaba. Design of high torque and high speed leg module for high power humanoid. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4497–4502, St. Louis, USA, Oct. 2010. IEEE. B.3.4

[96] Z. Vafa and S. Dubowsky. ON THE DYNAMICS OF MANIPULATORS IN SPACE USING THE VIRTUAL MANIPULATOR APPROACH. *IEEE International Conference on*, 1987. A.1

[97] R. Van Der Merwe, E. Wan, and S. Julier. Sigma-Point Kalman Filters for Nonlin-

ear Estimation and Sensor-Fusion. In *Proceedings of the AIAA Guidance, Navigation & Control Conference*, pages 5120–5159. Citeseer, 2004. 4.2.3

[98] G. Wahba. A Least Squares Estimate of Satellite Attitude. *SIAM Review*, 8(3):384–386, 1966. A.5

[99] S. Wakimoto, J. Nakajima, M. Takata, T. Kanda, and K. Suzumori. A micro snake-like robot for small pipe inspection. In *Micromechatronics and Human Science, 2003. MHS 2003. Proceedings of 2003 International Symposium on*, pages 303–308. IEEE, 2003. 2.4.3

[100] Z. Wang, Q. Cao, N. Luan, and L. Zhang. Development of an autonomous in-pipe robot for offshore pipeline maintenance. *Industrial Robot: An International Journal*, 37(2):177–184, 2010. 2.4.3

[101] M. M. Williamson. *Series Elastic Actuators*. PhD thesis, Massachusetts Institute of Technology, 1995. 2.4.1, 6.1

[102] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, D. Rollinson, M. Tesch, and H. Choset. Design and Architecture of the Unified Modular Snake Robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4347–4354, St. Paul, USA, 2012. 2.1, 2.2, 3, B

[103] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. L. Hatton, and H. Choset. Design of a Modular Snake Robot. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Oct. 2007. 2.1

[104] C. Wu and R. Paul. Manipulator Compliance Based on Joint Torque Control. *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 19(3):55–71, 1980. 2.4.1

[105] H. Yamada and S. Hirose. Study on the 3D shape of active cord mechanism. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2890–2895, Orlando, USA, 2006. IEEE. 2.3.2, 3.1.1, 5.1, 9.1

[106] H. Yamada and S. Hirose. Approximations to continuous curves of Active Cord Mechanism made of arc-shaped joints or double joints. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–708. IEEE, 2010. 2.1, 2.2

[107] H. Yamada and S. Hirose. Steering of pedal wave of a snake-like robot by superposition of curvatures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 419–424. IEEE, 2010. 2.1

[108] A. J. Yezzi. Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. *International Journal of Computer Vision*, 2003. A.1

[109] M. Yim, D. Duff, and K. Roufas. PolyBot: A Modular Reconfigurable Robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 514–520, San Francisco, USA, 2000. 2.1

[110] M. Yim, W. Shen, and B. Salemi. Modular self-reconfigurable robot systems.

Robotics & . . . , (March), 2007. 2.1

[111] P. Zarchan and H. Musoff. *Fundamentals of Kalman Filtering: A Practical Approach*. AIAA (American Institute of Aeronautics & Astronautics), 3rd editio edition, 2009. 3.1.2, 4.2.1