Spyndra 1.0: An Open-Source Proprioceptive Robot for Studies in Machine Self-Awareness

Ori Kedar[1], Christie Capper[1], Zhaoyang Chen[1], Julia Di[1], Yonah Elzora[1], Lingjian Kong[1], Julian Oks[1]
Jorge Orbay[1], Fabian Stute[1], Chad Tarpley[1], Joni Mici[1], and Hod Lipson[1]
[1] Columbia University, New York, NY 10027
oak2106@columbia.edu

Abstract

This paper describes *Spyndra*, a quadruped robot created as an open source platform for studying machine self-awareness. Our key hypothesis is that a machine is self-aware to the degree it is able to simulate itself, and that self-simulation is essentially the ability to predict sensations from actions. In order to facilitate studies in this direction, we create the first of what we see as a series of open platforms that provide rich proprioceptive feedback. This paper provides descriptions of Spyndra's hardware and software, as well as analysis of an Inertial Measurement Unit (IMU) dataset, which can serve as a baseline for future studies. All of Spyndra's parts are accessible, low cost and off-the-shelf or 3D-printable. Assembly instructions and software are open source.

Figure 1: Spyndra printed in a camouflaged brick pattern

Introduction

Questions over the nature of self-awareness and consciousness have occupied philosophers and physicians for millennia. The relatively recent advent of robotic systems, combined with machine learning technologies, has opened a new window into these age old questions. For the first time, some of the prevailing conjectures or models can be put to a test.

The key hypothesis we aim to study is the idea that consciousness, or self-awareness, is essentially the ability to self-simulate, or to perform 'mental time-travel,' and that emotions are essentially internal appraisals of that prediction.

A second aspect of our working hypothesis, is that self-awareness is not a black-or-white characteristic that creatures either possess or not. Alternatively, self-awareness lies on a continuum from machines with no self-awareness to human-level self-awareness, and beyond. If true, this hypothesis implies that we can begin experimenting with systems that might possess minute amounts of self-awareness.

Spyndra is a robotic platform that could potentially be capable of a small level of self-awareness. Utilizing minimal proprioceptive sensation to track its internal motor commands and record its own orientation and acceleration, Spyndra can learn about its physical form. We hope that by sharing this platform and data produced by it, researchers engaged in this line of research can study self-awareness on a common platform.

Growth within the robotics field has traditionally been limited by barriers to entry such as expensive components and complex, inaccessible manufacturing. By contrast, Spyndra was designed as an open source platform, using off-the-shelf electronics and 3D printed, easily-assembled hardware, with all necessary files and instructions available on the project's website. Table 1 provides a comparison of similar low cost walking robots, both academic and hobby in origin. Spyndra is comparable in price to these alternatives, at approximately $600 to build and operate. Furthermore, by limiting the DOF to eight, Spyndra reduces the complexity of potential gaits, making machine learning a more computationally tractable problem. Calibration procedures designed specifically for the robot ensure synchronicity between software and hardware and the repeatability of experiments. We have also developed control software for Spyndra, allowing gaits generated through machine learning to be seamlessly commanded to the robot, as well as a dataset of IMU readings that can serve as a baseline for further research. Though this paper presents Spyndra in its first version, the public availability of native files, including Python scripts and CAD files, allows for crowd-sourced customization and improvement of the system as it is adopted.

Robot	Image	Price	Open-source hardware	Sensors	”DoF” # and type
Spyndra [12]		$600	Yes: provides STLs and CAD	1 IMU, 1 camera	8 - rotary
Instructables Arduino Quadruped [13]		$540	Yes	Triple access accelerometer	12 - rotary
RobotShop Lynxmotion SQ3U Symmetric Quadruped Walking Robot [14]		$550	No: assembly kit	Not listed	12 - rotary
Instructables Spider Robot [15]		$\sim$$100	Yes: provides STLs, not CAD	Not listed	12 - rotary
Instructables 3D Printed Quadruped Robot [16]		$\sim$$450	Yes: provides STLs and CAD	Not listed	12 - rotary
Aracna [17]		$1,350	Yes: provides STLs and CAD	Not listed	8 - rotary
Hexy - Programmable Hexapod Kit [20]		$250	Yes: provides Code, Laser Cutter DXF/STL/CAD files	Ultrasonic Distance Sensor Eyes	18 -rotary
Lynxmotion A-Pod Hexapod [19]		$1499 (No electronics)	No: assembly kit	Force Sensors	18 -rotary

Table 1: Comparison of Spyndra to some other walking robots less than $1,500

Hardware

The hardware strikes a balance between being user-friendly to a wide audience and sophisticated enough to achieve a wide variety of tasks. The website includes a bill of materials, all native CAD and STL files, and instructions on how to fabricate and assemble the hardware, integrate the electronics, and implement the software. Spyndra's sensor systems include a camera and an Inertial Measurement Unit (IMU). These sensors provide information necessary for Spyndra to develop a model of itself and interact with its environment. The hardware can be customized to accommodate additional sensors.

For convention, this paper will refer to the upper section of each leg as the 'Femur,' and the lower section as the 'Tibia' as illustrated in Figure 2. A central chassis holds the Raspberry Pi 3B micro-controller, Lithium-ion battery (power source of the controller), and a Lithium polymer battery (power source for the servo motors), as well as several sensors. The chassis also houses four servos linked to the femur.

Unlike Spyndra's Creative Machines Lab predecessor, Aracna, the eight high-torque metal gear analog servo motors (Power HD 1501MG) directly drive each joint of the robot [5]. The two-pronged femurs support both sides of the motors to prevent load paths orthogonal to the motors intended axis of motion. Adhering to this design principle maximizes servo life and improves Spyndra's overall robustness. The motors slide effortlessly into place in both the chassis and tibia, and are connected to the femur using standard servo horns. This direct drive, as opposed to Aracna's linkage system, results in low friction/low hysteresis motion, that can be more accurately represented in simulation.

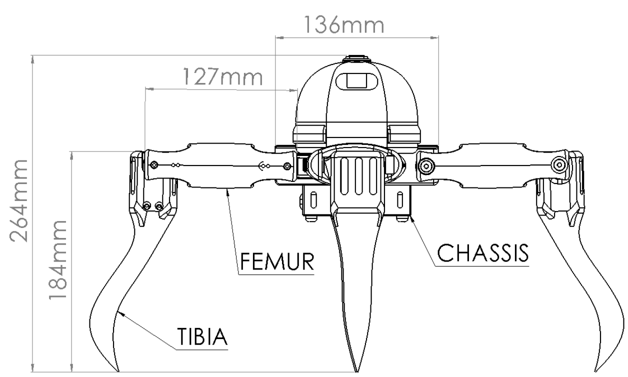


Figure 2: Labeled and dimensioned drawing of Spyndra

Spyndra is comprised of entirely 3D-printed parts, with all necessary STL files available on the project's website. The parts are nominally designed and tested for fabrication via Fused Filament Fabrication [9] using typically low cost desktop 3D printers (Ultimaker 2 Extended+) which enables the design of topologically complex bio-inspired parts.

	Robot Specifications
Diameter	Min: 7.75 in Standing: 10.75 in Max: 15.5 in
Height	Min: 5.5 in Standing: 10.75 in Max: 15.5 in
Weight	1.558 kg (+0.1kg with battery and voltage regulator)
Input Voltage	6V to motors 5V to Raspberry Pi
Input Amperage	Idle: 0.2A Standing: 2 A Walking: 6 A
Input Power	Idle: 1.2 W Standing: 12W Walking: 36W
Battery Run Time	Idle: 13.3 hours Standing: 72 minutes Walking: 24 minutes

Table 2: Physical and electrical specifications of Spyndra

The recommended materials are Polylactic Acid (PLA) and ABS plastics, which are inexpensive and allow for the simple application of heat-inserts for fastening. We also fabricated Spyndra models using a Stratsys J750 printer which uses PolyJet technology [9] to achieve full color and multi-material printing, as shown in Figure 1. The organic textures on these experimental models challenge the metallic motif which we have come to accept for modern day robots such as BigDog.

Various 3D printer settings, including wall-thickness, infill density, and layer deposition orientation, have been iteratively tested. Hardware failure was initially a recurrent issue with repeated use, however incidences of fracture have been reduced, almost to entirety, through minor design revisions and enhanced printer settings. Optimized settings and print orientations for Ultimaker 2.0 3D printers can be found on the project website.

Designed for ease of assembly, the components are either

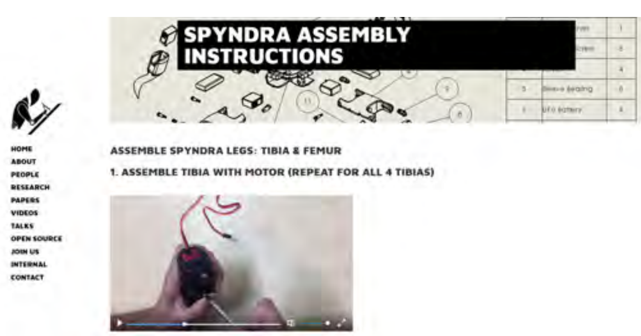


Figure 3: Parts list, software and assembly instructions are publicly available on the project website.

3D Printing Materials	$70
Controller	Raspberry Pi 3 + Adafruit Servo Hat: $55
Motors	8 x Power HD 1501MG: $160
Batteries	Li-Ion, LiPo, Battery Charger, Voltage Regulator: $115
Sensors	BNO055 IMU, Camera: $75
Misc. Electronics and Fasteners	$120

Table 3: Cost breakdown of Spyndra

press-fit or fastened using screws and heat-set inserts. No adhesives are needed in the the assembly of Spyndra. The only tools necessary to assemble Spyndra are a few screwdrivers and a soldering iron.

Powered by two batteries, Spyndra can function as an untethered robot. The Raspberry Pi 3 can either be programmed to execute autonomous programs upon booting, or receive commands wirelessly via USB, Bluetooth, or SSH protocol. Spyndra's present design can run for approximately twenty minutes on one charge, but extra Lithium polymer batteries for the servo motors can be added to increase lifespan.

Spyndra costs around $600 to build and operate. The price breakdown can be seen in Table 3. If printed with recommended settings using PLA filament, Spyndra weighs 1.55kg. However, further lightweighting and cost reduction can be accomplished by reducing infill settings, using lower torque motors, and higher strength printing materials. These changes may come at the expense of component robustness and lifespan.

To minimize mass, Spyndra uses a small Raspberry Pi-compatible camera. The camera interfaces with Spyndra's software by using the Raspicam commands native to the Raspberry Pi. The visual information from the camera, coupled with deep learning networks, offer a multitude of capabilities, including object recognition and the ability to obtain depth information. The camera has so far been used for image recognition, with the ultimate aim of finding waypoints for path planning.

Adafruit's BNO055 is the Inertial Measurement Unit (IMU) used for measurements of acceleration, rotation, and magnetic orientation each along three dimensions and provides primary feedback for gait generation. Using integration, the IMU can provide a variety of information about Spyndra's physical state, such as its position, velocity, and acceleration. Since the Spyndra architecture exhibits two-way symmetry, the IMU is placed in the geometric center of the chassis so it collects the most inertially relevant data. The output from the IMU can be used to train Spyndra's algorithms to improve its ability to learn to walk.

Gait Generation Software

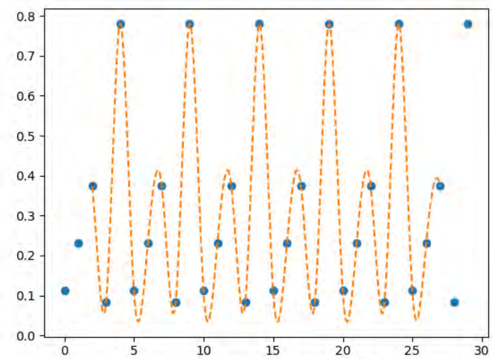
Spyndra's software suite currently consists three spline generation scripts and two runner scripts. The first spline generator is a random spline generator, which creates two random arrays of five points each, one array corresponding to femur position, and the other tibia position. The generator then fits the arrays to splines, and outputs a large array of percentages to be mapped to motor angles, see Figure 4. The five auto generated points are also tested for high deviation, and points too far away from each other are re-generated to avoid erratic movements which can damage the servo motors. This process results in a "random gait," which will be realized on the hardware using a runner script. The second spline generator produces what we call a "standing gait." It works similarly to the first but always produces the same motor angles, in which Spyndra has no translational movement, but only gyrates the chassis. This gait is meant for IMU calibration and testing. The third spline generator produces a "manual gait." The user inputs the desired number of motor coordinates for Spyndra's femur and tibia joints, and the generator fits these motor coordinates to a spline which is then outputted as an array of motor angles. For all three gait generators, the user can decide the number of times the spline is repeated.

When producing a gait, the relevant information can be stored into a JSON file for later use. An example is show below in Listing 1.

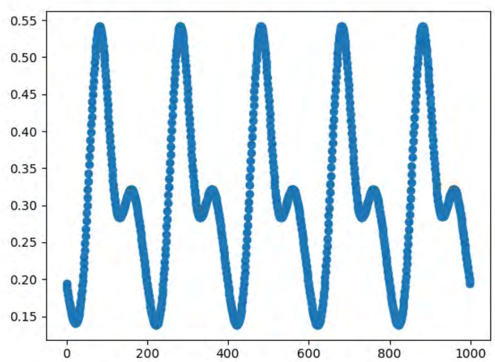
Listing 1: JSON storage of Gait

```
{"Gait": {
            "Title": "Standing Gait",
            "Femur IDs": [1,3,5,7],
            "Tibia IDs": [2,4,6,8],
            "Femur Sequence":
            [100,120,140,160,180],
            "Tibia Sequence":
            [260,240,220,200,180],
            "Offset": 45
        }
}
```

The first runner script takes the output of any of the three generators and parses the splines, with each leg running the same spline. First, the program prompts the user for the desired phase offset, which is the amount of lag between legs as they run the spline. The spline, which consists of an array of percentages, is then mapped to the appropriate maximum and minimum motor angles (interpreted as PWM signals) which are referenced from a calibration log file. The PWM signals are sent to the servo motors while the IMU data is logged in a separate file. The second runner script is identical to the first, but accepts a time offset between legs, as opposed to a phase angle.



(a)



(b)

Figure 4: Random points generated for gait are a) fit to spline, then b) the spline is sampled to produce the gait array

Before executing the gait, both runner scripts first slowly move Spyndra to a standing position with the femurs parallel and the tibias perpendicular to the robot's chassis. Once Spyndra is standing, the designated spline runs for the desired amount of loops. Finally, once the spline has finished running, Spyndra is moved to a sitting position where the tibia and femur are outstretched and the robot is resting on its chassis. Upon completion, any random gait generated has the option to be saved to a log file if desired.

Calibration

To ensure the repeatability of gaits, proper calibration of Spyndra is key. When properly calibrated, all four of Spyndra's legs will go to the same physical position when given the same PWM command, as illustrated in Figure 5. The robot consists of 8 servo motor actuators that are connected to the legs using servo horns (as discussed in Section : Hardware). The servos have a travel limit of roughly 165 degrees, and the horns must be attached in the proper orientation. To ensure that all limbs are properly calibrated, we have engineered a combined mechanical and software approach.

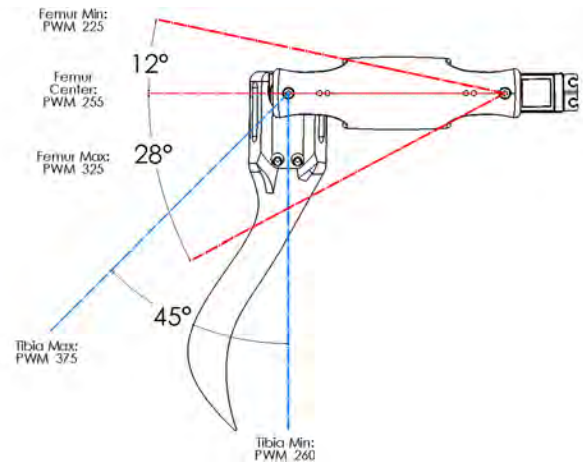


Figure 5: Range of motion of tibia and femur with corresponding PWM signals

First, the motors are unplugged and shifted all the way back to the maximum angle (counter clockwise for tibia, clockwise for femur). A mechanical jig is then placed and used to attach the horn to each legs servo motor as shown in Figure 6. This provide a rough calibration, but the coupling mechanism of the horn limits the precision of each legs orientation to within 14 degrees.

Second, to further refine Spyndra's leg calibration, we lay Spyndra on a flat surface and raise its femurs and tibias to their maximum positions. Then, one at a time, each femur is lowered until it touches the flat surface. The angle at which contact occurs is recorded for each individual femur, and these angles are used to define the range of motion of each femur. The femurs are then raised ten degrees above contact, and the same process is done for the tibias. The low-

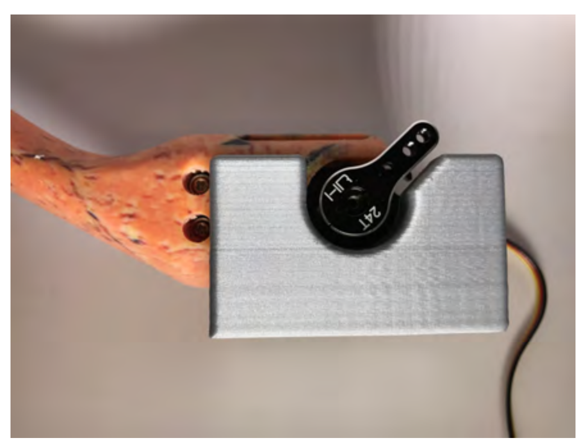
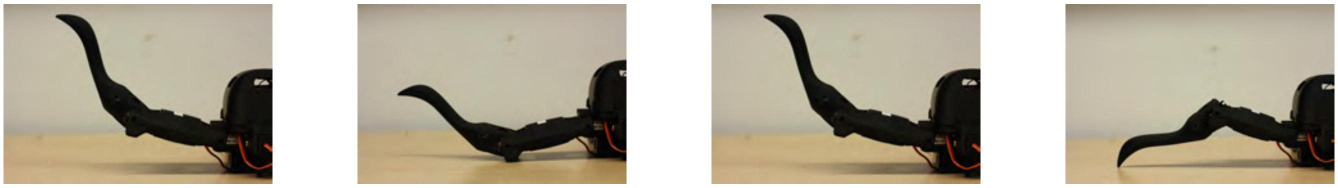


Figure 6: The 3D printed mechanical jig (grey) is placed on the motor and guides the coupling of the servo horn (black) to the proper orientation.



(a) The sequence for each leg starts with the Spyndra on a flat surface. The femur and tibia are raised to their maximum angles.

(b) The femur is lowered until it touches the flat surface at its lowest point. The PWM period of this point is recorded.

(c) The femur's position is raised twelve degrees from the position of contact.

(d) Now, the tibia is lowered until it touches the flat surface at its lowest point. The PWM period of this point is recorded.

Figure 7: The calibration sequence for a single leg

est increment of angle is about 0.4 degrees, so all the legs are calibrated within 0.4 degrees of each other. This allows users to design gaits that can treat each leg as equivalent to the others. Figure 7 demonstrates this process.

IMU Dataset

To demonstrate Spyndra's suitability as a robotic gait machine learning platform, two datasets were taken using IMU measurements. These datasets demonstrate the repeatability of measurements and can serve as a baseline for future gait studies.

Both datasets consist of time, yaw, pitch, roll, x-acceleration, y-acceleration, and z-acceleration data from Spyndra. In the first, Spyndra performs the "standing gait" (described in the Section: Gait Generation Software), with a phase offset of 90 degrees. In this "standing gait Spyndra

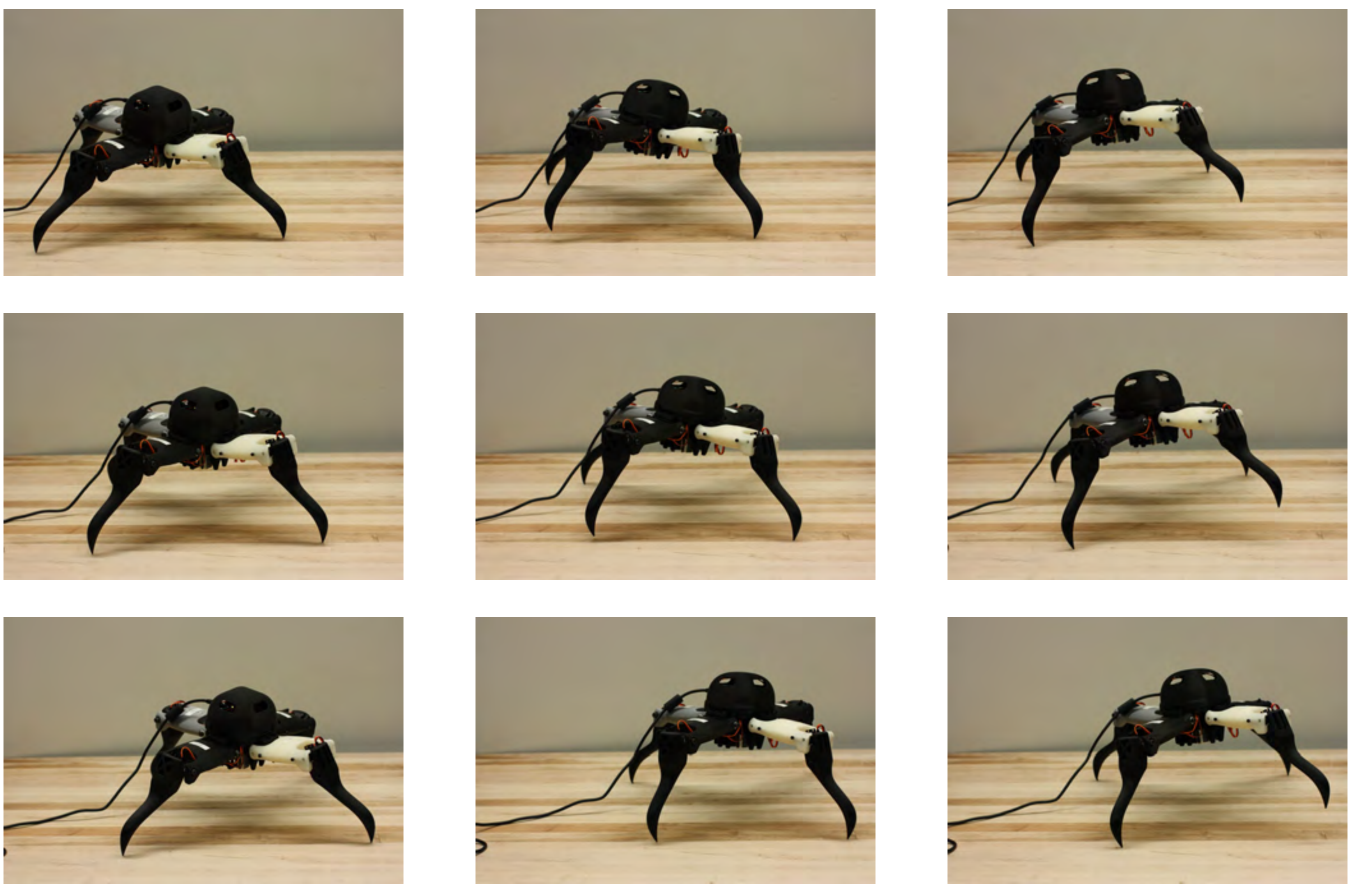
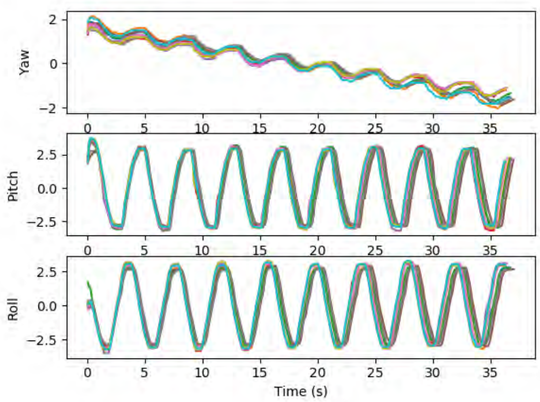
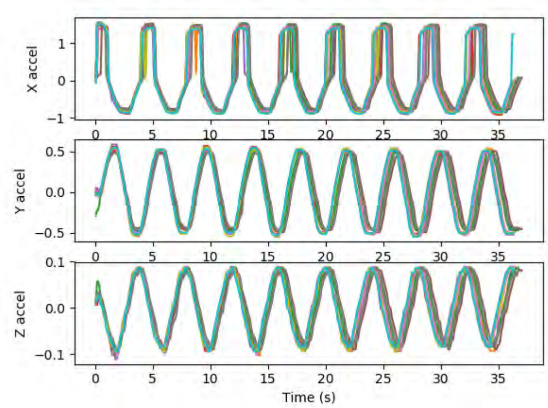


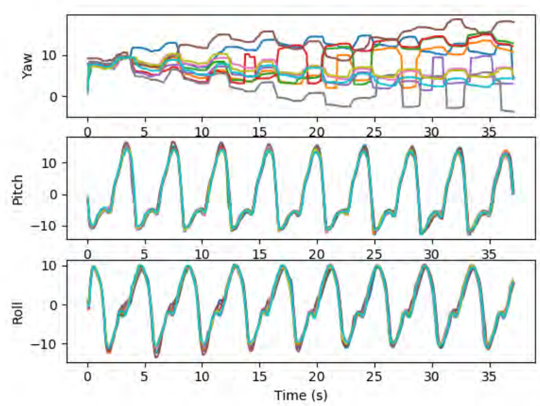
Figure 8: A frame by frame illustration of Spyndra's "walking gait" measured for IMU dataset. Camera position is fixed.



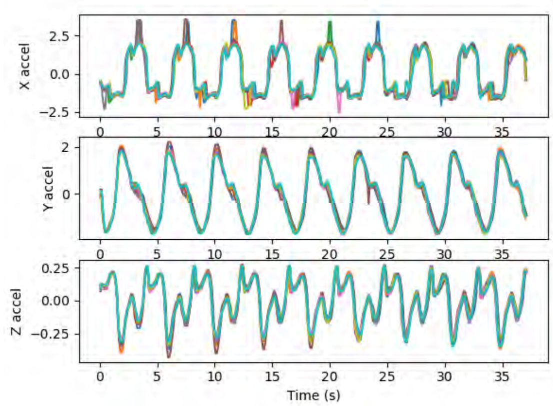
(a) Gyroscopic data from 10 repetitions of standing gait.



(b) Accelerometer data from 10 repetitions of standing gait.



(c) Gyroscopic data from 10 repetitions of walking gait.



(d) Accelerometer data from 10 repetitions of walking gait.

Figure 9: IMU data plotted against time. Gyroscopic data in degrees, acceleration data in m/s^2

	Mean Correlation Coefficient	**Standard Deviation of Correlation Coefficients**
Yaw	0.9645	0.0286
Roll	0.9972	0.0016
Pitch	0.9959	0.0018
X Acceleration	0.9451	0.0236
Y Acceleration	0.9971	0.0016
Z Acceleration	0.9925	0.0028

(a) Mean and standard deviation of correlation coefficients across repetitions for standing gait.

	Mean Correlation Coefficient	**Standard Deviation of Correlation Coefficients**
Yaw	0.2586	0.4832
Roll	0.9917	0.0048
Pitch	0.9952	0.0022
X Acceleration	0.958	0.0129
Y Acceleration	0.9919	0.0047
Z Acceleration	0.9823	0.0072

(b) Mean and standard deviation of correlation coefficients across repetitions for walking gait.

Table 4: Distribution of correlation coefficients of IMU data across repetitions

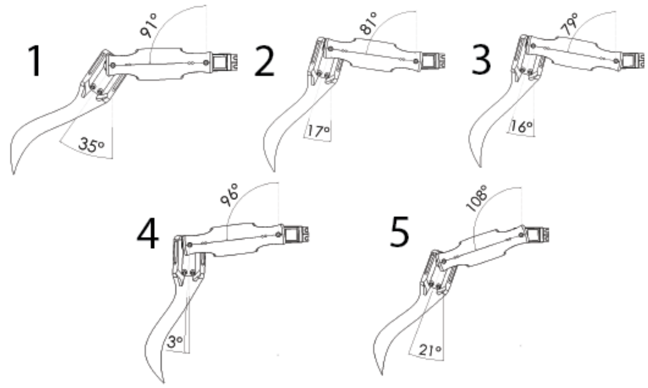


Figure 10: Illustration of the "walking gait" performed by Spyndra for data collection. An offset of 45 degrees was used between legs.

slowly gyrates the chassis while standing stationary.

The second dataset was recorded as Spyndra performed a stored randomly generated "walking gait," with the pattern shown in Figure 4a. A phase offset of 45 degrees was used between legs. As friction of the walking surface affects Spyndra's gait, it is important to note that the experiment was conducted on a linoleum floor. Both datasets include data from 10 gaits each.

For both experiments, standard hardware was used, and Spyndra's motors were powered using a 5V 10A max power adapter (rather than by LiPo battery). The raw data can be found on the Spyndra website as well as the Python script used to process it. Data processing included filtering of anomalous high and low values, use of a median filter to remove sharp spikes in data, and normalizing of data to account for differences in initial orientation.

Analysis of "standing gait" data reveals high correlation between tests, as seen in the plots in Figure 9a and b. To quantify this correlation, we calculated correlation coefficients between tests, and display the average values and standard deviations across repetitions in Table 4a. Oscillation of gyroscopic and acceleration data values are explained by the cyclic gyrating motion of the chassis during this routine. Note that there is a cumulative yaw rotation during the routine.

Analysis of the "walking gait" also reveals high correlation of results in most data as seen in the graphs in Figure 9c and d and the correlation coefficients in Table 4b. Yaw data, however, for this test had very low correlation between tests and is too noisy to determine any recognizable patterns. Again, the oscillation of gyroscopic and acceleration data are caused by the cyclic pattern of a repeated gait.

The high correlation between data from repeated gaits, demonstrate Spyndra's suitability as a platform for gait generation using machine learning.

Conclusion

We have introduced Spyndra: an open source quadruped robot meant to serve as a platform for robotics and AI researchers interested in self-awareness. Comprised of 3D-printed parts and off the shelf hardware, Spyndra is inexpensive, easy to assemble, yet achieves complex kinematics. To aid in the implementation of machine learning, open source control software is available, and a set of baseline IMU data are available for future researchers. These materials make Spyndra ideal for hardware implementation of machine learning and self-modelling software and we hope it will serve as a common starting point among roboticists, academics, and the broader AI community.

In the future, we would like to strengthen Spyndra's capabilities as a self-modeling platform by increasing the feedback sensors on board. With additional proprioceptive sensors, Spyndra can learn more about itself. In addition we will collect more data from the IMU, growing the publicly available dataset, and creating an accurate simulated model of the hardware.

Acknowledgements

This work has been supported in part by a gift from Northrop Grumman Incorporated, for the study of self-aware systems.

References

[1] Bongard J., Zykov V, Lipson H. (2006). "Resilient machines through continuous self-modeling." *Science*, 314(5802):1118-21.

[2] Hornby, G., Takamura, S., Yamamoto, T., and Fujita, M. (2005). Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, 21(3):402410.

[3] Koza, J. (2003). Genetic programming IV: Routine human competitive machine intelligence. Kluwer.

[4] Lipson, H. and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. Nature, 406(6799):974978.

[5] Lohmann, Sara, Jason Yosinski, Eric Gold, Jeff Clune, Jeremy Blum, and Hod Lipson. "Aracna: An Open-Source Quadruped Platform for Evolutionary Robotics." Artificial Life 13 (2012): n. pag. Web.

[6] Mnih, V. et al. Human-level control through deep reinforcement learning. Nature 518, 529533 (2015).

[7] Nolfi, S. and Floreano, D. (2000). Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. MIT Press, Cambridge, MA.

[8] Pfeifer, R., Bongard, J., and Grand, S. (2007). How the body shapes the way we think: a new view of intelligence. The MIT Press.

[9] Standard, A. S. T. M. "F2792. 2012 Standard terminology for additive manufacturing technologies." West Conshohocken, PA: ASTM International. doi: 10.1520/F2792-12

[10] Sims, K. (1994). Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4):353-372

[11] Yosinski J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J., and Lipson, H. (2011). Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization. *In Proceedings of the European Conference on Artificial Life*, pages 890-897

[12] *Spyndra* [Online] Creative Machines Lab - Columbia University. [Online]. Available: http://www.creativemachineslab.com/spyndra.html [7 Apr. 2017]

[13] Hend, D.*Arduino Quadruped Robot* Instructables. [Online]. Available: http://www.instructables.com/id/Synopsis/[7 Apr. 2017]

[14] *Lynxmotion SQ3U Walking Robot* Lynxmotion. [Online]. Available: http://www.lynxmotion.com/c-26-quadrapods.aspx [7 Apr. 2017]

[15] Hsu, R.*[DIY] Spider Robot(Quad Robot, Quadruped)* Instructables. [Online]. http://www.instructables.com/id/DIY-Spider-RobotQuad-robot-Quadruped/[7 Apr. 2017]

[16] Hatfield, S.*A 3D Printed Quadruped Robot* Instructables. [Online]. http://www.instructables.com/id/A-3D-Printed-Quadruped-Robot/[7 Apr. 2017]

[17] *Aracna* Creative Machines Lab - Columbia University. [Online]. Available: http://www.creativemachineslab.com/aracna-robot.html [7 Apr. 2017]

[18] Bongard, J. *Starfish* Mechanical and Aerospace Engineering - Cornell University. [Online]. Available: http://www.pnas.org/content/108/4/1234.abstract [7 Apr. 2017]

[19] *Lynxmotion A-Pod Hexapod* Lynxmotion. [Online]. Available: http://www.robotshop.com/en/lynxmotion-a-pod-hexapod-robot-kit-no-electronics.htmlSpecifications [15 Apr. 2017]

[20] *Hexy-Programmable Hexapod Kit* Adafruit. [Online]. Available: https://www.adafruit.com/product/1529 [15 Apr. 2017]

Time	**Yaw**	**Pitch**	**Roll**	**Acc**$_X$	**Acc**$_Y$	**Acc**$_Z$	**M0**	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**
0.00	1.13	-1.81	-9.25	-0.31	1.58	9.67	254	348	326	334	283	286	283	366
0.11	1689	2035	-1.06	-0.62	1.53	-319	252	341	326	334	289	286	276	366
0.17	1690	2032	2041	-1.30	-325	-319	250	333	324	335	295	288	270	364
0.23	1690	-8.69	-3.94	-1.49	-325	-319	250	325	321	337	299	293	264	360
0.31	4.81	-9.56	-1.00	-0.35	0.17	9.66	252	316	318	341	303	299	259	355
0.41	4.69	-9.69	-0.13	-0.37	0.02	9.66	255	309	313	346	307	306	255	348
0.51	4.69	-9.06	-0.13	-0.27	0.02	9.68	259	302	308	351	310	313	252	341
0.62	4.63	-8.75	-0.63	-0.21	0.11	9.69	265	296	302	356	313	320	250	333
0.72	4.56	-8.75	-1.38	-0.23	0.23	9.68	271	291	295	361	315	326	250	325
0.82	4.56	-7.31	-1.63	-1.25	0.28	9.72	278	287	289	364	317	331	251	317
0.92	4.50	-6.25	-2.06	-1.07	0.35	9.74	284	286	282	366	320	334	255	309
1.02	4.38	-5.63	-2.88	-0.96	0.49	9.74	290	286	276	366	322	336	259	302
1.13	3.69	-4.75	-4.38	-0.82	0.75	9.74	295	289	269	364	324	336	265	296
1.23	4.19	-4.50	-7.19	-0.77	1.22	9.69	300	293	264	360	325	335	271	291
1.33	4.06	-4.50	-9.75	-0.77	1.66	9.63	304	299	259	355	326	334	277	287
1.43	3.81	-4.50	-13.3	-0.77	2.26	9.50	307	306	254	348	326	334	283	286
1.54	4.44	-3.81	-19.1	-0.65	3.22	9.23	310	313	252	341	326	334	289	286
1.64	3.31	-3.44	-21.5	-0.59	3.57	9.11	313	320	250	333	324	335	295	288
1.74	3.38	-2.75	-22.3	-0.47	3.72	9.05	315	327	250	325	321	337	299	293
1.84	3.81	-1.13	-22.2	-0.19	3.71	9.07	317	332	252	316	318	341	303	299
1.95	4.19	0.31	-21.6	0.05	3.62	9.11	320	335	255	309	313	346	307	306
2.05	4.25	0.63	-20.4	0.10	3.44	9.18	322	336	259	302	308	351	310	313
2.15	4.88	1.44	-17.7	1.53	2.99	9.33	324	336	265	296	302	356	313	320
2.26	5.19	2.38	-14.3	1.69	2.44	9.48	325	335	271	291	295	361	315	326
2.36	5.44	3.19	-13.2	1.82	2.24	9.52	326	334	278	287	289	364	317	331
2.46	5.63	4.19	-12.6	1.99	2.14	9.54	326	334	284	286	282	366	320	334
2.56	5.88	5.44	-10.8	2.21	1.84	9.58	326	334	290	286	276	366	322	336
2.66	5.94	6.69	-10.2	2.41	1.74	9.58	324	335	295	289	269	364	324	336
2.76	6.00	7.31	-9.8	2.53	1.67	9.58	321	337	300	293	264	360	325	335
2.87	6.06	9.38	-9.8	1.60	1.66	9.53	317	341	304	299	259	355	326	334
2.97	5.94	10.50	-10.3	1.78	1.74	9.48	313	346	307	306	254	348	326	334
3.07	2.19	9.94	-10.8	1.69	1.82	9.48	307	352	310	313	252	341	326	334
3.17	5.81	10.81	-11.0	1.84	1.85	9.45	301	357	313	320	250	333	324	335
3.28	2.06	11.44	-10.1	1.95	1.69	9.46	295	361	315	327	250	325	321	337
3.38	6.00	11.25	-9.2	1.92	1.55	9.49	288	365	317	332	252	316	318	341
3.48	6.13	10.56	-7.6	1.81	1.32	9.54	282	366	320	335	255	309	313	346
3.58	6.25	10.00	-6.1	1.70	1.03	9.60	275	366	322	336	259	302	308	351
3.69	2.69	7.56	-5.7	1.29	0.97	9.67	269	364	324	336	265	296	302	356
3.79	4.13	3.13	-4.8	1.88	0.84	9.75	263	360	325	335	271	291	295	361
3.89	2.94	-1.19	-3.5	-0.21	0.60	9.78	258	354	326	334	278	287	289	364
4.00	2.19	-5.81	-2.6	-1.00	0.45	9.74	254	348	326	334	284	286	282	366
4.10	1.56	-10.2	-2.1	-0.47	0.37	9.64	251	340	326	334	290	286	276	366
4.20	5.88	-9.56	-1.4	-1.63	0.24	9.66	250	332	324	335	295	289	269	364
4.30	2.06	-9.75	-1.0	-0.38	0.18	9.66	250	324	321	337	300	293	264	360
4.41	2.06	-9.75	-0.4	-0.38	0.07	9.66	252	316	317	341	304	299	259	355
4.51	2.06	-9.25	0.00	-0.30	0.00	9.67	255	308	313	346	307	306	254	348
4.61	2.06	-8.56	0.00	-0.18	0.00	9.69	260	301	307	352	310	313	252	341
4.71	2.00	-8.44	-0.8	-0.16	0.15	9.69	265	295	301	357	313	320	250	333
4.81	1.94	-7.81	-1.7	-0.07	0.29	9.70	272	290	295	361	315	327	250	325
4.92	1.88	-7.13	-1.9	-1.22	0.33	9.72	278	287	288	365	317	332	252	316
5.02	1.81	-6.25	-2.5	-1.07	0.43	9.73	284	286	282	366	320	335	255	309
5.12	1.69	-6.19	-3.5	-1.06	0.61	9.72	290	286	275	366	322	336	259	302
5.22	1.63	-5.69	-5.2	-0.98	0.89	9.71	295	289	269	364	324	336	265	296

Appendix 1: Example of format of IMU data found on website. Column 1: time (s), Columns 2-4: gyroscopic data (degrees), Columns 5-7: accelerations (m/s^2), Columns 8-15: motor commands (PWM signal)