Digital Servo Calibration and Modeling

Ethan Tira-Thompson

***Abstract*—The introduction of digital microcontrollers into "hobby" servos opens new doors for consumer and educational robotics. However, the new operational modes, parameters, and sensory feedback also add complexity. This paper will analyze the capabilities of these servos, and describe methods of calibration and motion modeling for accurate planning and control. As much as possible, these methods avoid use of precision rigs or expensive measurement devices to remain accessible to the classrooms, laboratories, and garages which these servos target.**

I. Survey of Servo Models and Applications

RC servos or "hobbyist" servos are typically intended for use in small remotely controlled devices such as model boats or planes. Due to their standard control interface, low cost, and wide variety of sizes, torques, and materials, these servos have become a common component of indoor robotics, particularly among prototype and research oriented devices.

Unfortunately, their association with teleoperated devices with limited range of motion and small number of actuators implies several design choices which constrain autonomous capabilities, and are less than ideal for robotics in general. The lack of middle ground between RC servos and more fully-featured industrial counterparts has driven the creation of projects such as OpenServo [1] and other efforts [2] to replace standard RC servos brains with custom electronics.

The large manufacturers (e.g. Hitec, Futaba) have begun to address the robotics market by "digital servos", providing better response time and torque management. However, the "digital" refers to a microcontroller within the servo, not the communication method, which is still the same analog pulse-width modulation (PWM) interface used by hobby servos.

The restriction to analog communication is at the root of a number of problems:

- Unknown position - lack of feedback during motion is inconvenient, but startup is perilous as the servo snaps directly to its first command regardless of obstacles.
- Unknown load - users cannot detect collisions or measure weight and pressure distribution.
- Static controller parameters - different tasks may require different levels of compliance, speed, or other operational modes.
- Cabling - a well actuated manipulator requires many wires to carry the control signals out to each servo, increasing weight and snagging or pinching of wires.

This work supported by National Science Foundation award DUE-0717705
E. Tira-Thompson is a graduate student at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 (ejt@cs.cmu.edu)

Some of Hitec's HSR series of servos (HSR-5990, HSR-8498) do provide communication options beyond PWM. Designated as Hitec Multi-protocol Interface (HMI), this allows the same cable to be used as a conventional PWM interface, or a 19200 baud serial-TTL line.

The PWM interface is bidirectional, where a set of special pulse lengths can request a pulse-width response to indicate the servo's current position, or to switch between different pre-defined control parameter sets. However, servo controllers do not typically have sensing circuitry on the servo pins, and are not able to read a return pulse width. Further, the manufacturer's documentation notes: "Because the positional feedback [...] operates in conjunction with the PWM control function, there is a chance that a communication error will occur 10% of the time." [3] The PWM read signal also interrupts torque control, putting the servo into a back-drivable state until the next position command is received.

The serial-TTL interface, on the other hand, is more promising for flexible digital communication, but it is currently not well documented and almost entirely unpublicized. For example, some servo specification sheets list PWM as the only communication interface, even if it may be noted elsewhere that the servo supports HMI.

One attractive alternative is the Dynamixel line of servos from Robotis, which has a well documented protocol and a 1 megabaud serial TTL connection. These servos provide comparable torque, and provide speed, load, and position feedback, yet are priced lower than Hitec's. Controller parameters can also be configured. The servos allow 300° range of travel for position control, and can be switched into a "free spin" mode for continuous rotation under speed control, which can pass through the 60° dead zone.

Due to these advantages, the Dynamixel servo was selected for further study as to its feasibility for precision control and dynamic operation.

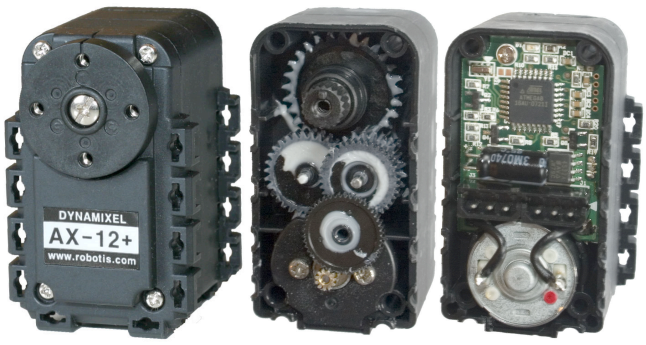

Fig. 1: The Dynamixel AX-12+ from Robotis; Exterior front, interior front, and interior back

II. Dynamixel Features

	7 volt	10 volt
Holding Torque	12.0 kgf·cm	16.5 kgf·cm
No-Load Speed	0.269 sec / 60°	0.196 sec / 60°

Table 1: published specifications for the AX-12

This paper is based on use of the power supply included with the Bioloid kit, which is rated at 12 volts (measured by the servos' own feedback as 12.3 volts). Different supply voltages will yield different servo responses, an effect which is not quantified here.

The servos provide feedback on position, speed, load, voltage, and temperature. Only the first three parameters will be evaluated in depth in the next section.

The servos' controller exports parameters for torque limit, punch, and separate clockwise/counter-clockwise compliance margin and slope. The 'margin' parameter controls how much error the servo will allow, the 'slope' specifies the proportional gain to reduce error, and the 'punch' provides an initial kick once the margin is exceeded.

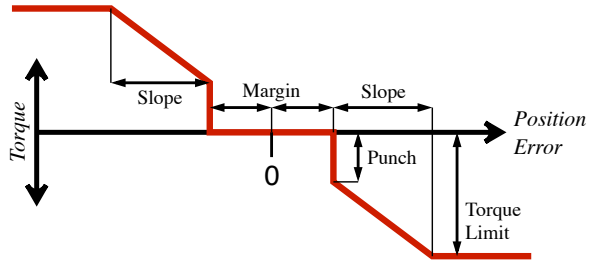

Fig. 2: Dynamixel controller model

Each servo is identified by a number between 0 and 253. 254 is used as a broadcast address. Commands are sent by writing values into enumerated registers, such as register 25 to control the integrated LED, or registers 30-31 for the goal position. (Thus, some parameters are 2-byte values, although the valid range is limited to [0,1023].)

There is a variety of commands for writing to the registers, providing features such as buffered updates for synchronized motion. To read sensor values back, each servo must be polled individually.

III. Goals and Related Work

The servo's position, speed, and load parameters are of particular interest, but only the position has a defined mapping to physical units from the manufacturer. (0–1023 maps to 0–300°).

The speed parameter presents a minor complication, because it is not consistent between any of commanded speed, physically measured speed, and reported speed, although each exhibits a straightforward linear correlation with the others.

The load parameter is more complicated. An accurate model of the load parameter profile during unloaded motion would be instrumental in allowing us to detect external contacts and forces. This type of sensing has proven very helpful in our experience on the Sony Aibo, where duty cycle information from the servos was a relatively clean signal and good indication of end-effector forces.

Some previous work in improving the accuracy of servos is presented in [4]. However it focuses on optimizing repetitive tasks, whereas this work is focused on generalized modeling the servo itself, applied to novel tasks.

On the other end of spectrum, there is a large body of work (e.g. [5]) regarding control of direct-drive motors in torque space, but these are typically large, powerful motors with low (or no) gear ratios. The dynamics of these systems degrade with large gear trains found in small servo motors, and does not generalize well to this hardware.

IV. Position Calibration

In order to verify servo position, a marker is attached to the servo and its orientation measured by computer vision.

A rough calibration target is used, explicitly avoiding any reliance on precise construction. Here, a simple piece of cardboard with a thin stripe is used. (Fig. 3) The stripe orientation can be easily detected using the DualCoding module of the Tekkotsu framework. [6]

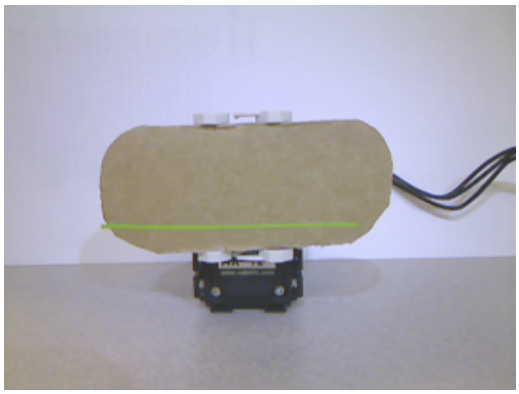

Fig. 3: Servo with a haphazard strip of green tape mounted to the horn. It is a good idea to clamp or bolt the servo to the table.

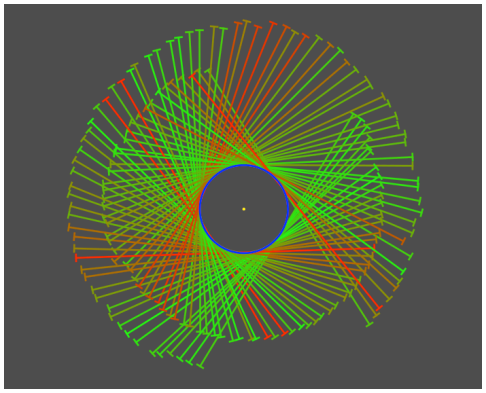

Fig. 4: Line samples, green-red color indicates variance from predicted center of rotation (green being lowest variance). Rotation center is consistent within a standard deviation (σ) of 0.21 pixels.

The accuracy of the camera's measurements is verified by a series of physically measured positions. The deviation of these values from the camera measurement was less than the

resolution of the physical measurements (1°) indicating that the visual measurement is consistent with ground truth.

Additional automated tests using the camera are then performed to verify servo position accuracy across the entire range of motion for each of seven servos, measuring inter-servo variance.

Servo #	Fit Slope $P_{vis} = x \cdot P_{Fdbk}$	Fit σ (°)	Target σ (°)
1	0.9992	0.42	0.30
2	0.9980	0.60	0.30
3	0.9996	0.44	0.31
4	1.0003	1.63	0.36
5	1.0101	1.33	0.35
6	0.9994	0.52	0.34
7	1.0037	0.48	0.31
Aggregate	1.0014	1.39	0.36
$\mu_x = 1.0015,\ \sigma_x = 0.0042$			

Table 2: Results for each of seven servos, linear fit of position feedback to visually measured position, with standard deviation from that fit, and the deviation of feedback position from target position. The first three servos are from a single Bioloids kit, whereas the remaining four were purchased as individual units.

The linear least squares fit of the aggregate feedback values to visually measured positions indicates that the average uncalibrated servo has a position bias within ±0.21° at the ends of its range of motion. This is less than the verified accuracy of the camera measurements and less than the servo's own resolution, indicating there is no statistical nor practical basis for applying a global positional calibration.

The overall standard deviation of position feedback from target position is 0.36°, corresponding to 1.21 units of servo resolution, and demonstrating the servo accurately reaches its target position when unloaded.

V. Speed Calibration

Once the Dynamixel servo is placed into "free spin" mode, it can produce continuous rotation, passing through the 60° dead-zone. As the servo passes through this dead-zone, the position feedback reports the closer of the minimum or maximum value, except in the middle of the dead-zone, where it reports a not quite constant mid-range value.

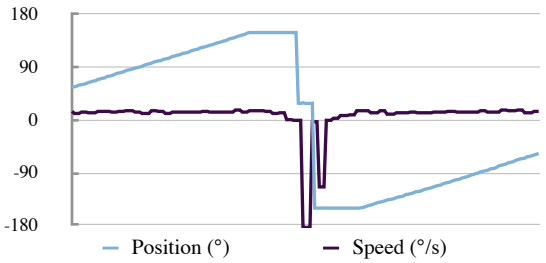

Fig. 5: Position and speed during dead-zone traversal by free spin

The speed feedback continues reporting values normally until the point at which these mid-range values are returned, where it leaps between a large value in the opposite direction of motion and zero, before resuming normal operation.

Speed feedback is updated on the order of every 130 milliseconds. It is possible to poll the servo at a much higher rate and get current position data each time, but the speed register holds constant between each of the internal updates.

To calibrate the commanded speed and corresponding feedback to physical units, a servo is commanded to a random speed, given 0.4 seconds to accelerate, and then its average feedback and physical displacement are recorded for the following 1.5 seconds. The servo is then brought to a full stop, and the process is repeated. If the servo enters the dead zone, the sample is stopped early; if less than half of the run time has passed, the sample is discarded.

The results for a single servo are plotted in Fig. 6, and the calibration parameters for a set of servos is in Table 3.

Of note in Fig. 6 is the horizontal plateau of commanded speeds with zero motion output, most likely due to static gear friction preventing any motion below a minimum torque. This indicates the servo is not performing integrative control of the speed, as it will not ramp up torque to maintain speed. This can be confirmed by manually holding the servo horn during a slow rotation, where the servo is easily immobilized and does not attempt to increase its torque.

This presents a problem for slow rotations, even after a linear calibration is performed as shown in Fig. 7. Slow rotations are limited to very low torque, and are not reliably produced as small perturbations can halt the servo mid-rotation. This yields the anomaly seen in the center of Fig. 7, where some samples are stuck at zero radians per second, and other samples are able to move, although both are given the same target speed.

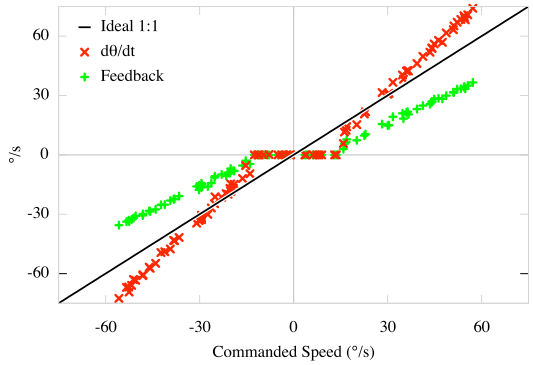

Fig. 6: Uncalibrated command vs. produced speeds

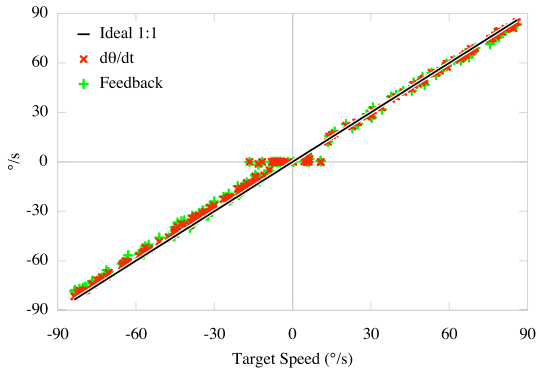

Fig. 7: Calibrated command vs. produced speeds

The "punch" parameter would ideally be set to counter static gear friction, but trials indicate none of the controller parameters (slope, punch, margin) affect free spin mode.

Free spin mode may be more useful for torque control than speed control. For slow rotations which do not need to pass through the dead zone, consecutive high-frequency position commands will more reliably produce a desired speed invariant to gravity and external loads.

Servo #	Fit: $S_{cmd} = x \cdot S_{tgt} + sign(S_{tgt}) \cdot b$		Target σ (°/sec)
	x	b	
1	0.728	7.506	0.859
2	0.695	9.454	0.917
3	0.671	7.907	0.974
4	0.776	14.381	3.610
5	0.764	15.699	3.839
6	0.742	11.517	1.604
7	0.682	14.954	1.662
Aggregate	0.705	12.834	5.959
μ of Fits:	0.722	11.631	
σ of Fits	0.041	3.438	

Table 3: Speed command calibration results for the same seven servos shown previously in table 2.

Although the aggregate fit provides a ballpark default calibration suitable for general tasks, applying separate calibration parameters to individual servos will significantly reduce the deviation of the speed error beyond that provided by the aggregate fit.

On the other hand, the results for speed feedback (Table 4) are much more consistent between servo units, encouraging a straightforward global calibration.

Servo #	Fit Slope $S_{actual} = x \cdot S_{Fdbk}$	Fit σ (rad/sec)
1	2.084	1.4152
2	2.066	1.8106
3	2.051	1.6215
4	2.074	1.9653
5	2.068	2.1543
6	2.080	1.9080
7	2.050	1.6730
Aggregate	2.066	1.8965
μ of Fits:	2.0675	
σ of Fits	0.0133	

Table 4: Speed feedback calibration results. Aggregate fit standard deviation is on par with individual unit fit deviations.

VI. Load Calibration

There are multiple goals for understanding how the servo measures and applies torque:

1) Anticipate and counter positional deflection
2) Map between free-spin "speed" and applied torque
3) Gauge external loads, e.g. objects in the gripper

As an example of how these would be useful, imagine launching a snowball. Each snowball is slightly different, and must be weighed to predict its trajectory. Throwing the snowball requires accurate dynamics, and cannot be practiced. (You can only throw a snowball once!)

To measure and model these load related issues, a rigid metal bar is affixed to the servo, providing attachments at known radii. The servo is clamped to a table corner, where the bar and attached masses can hang over the side. (Fig. 8)

Several motions are produced, in each case raising the bar to 45° above the horizontal and then lowering back to the table. (Fig. 9) Only data where the servo is in motion is used, so portions where the servo has not yet lifted from the table and where it is at its apex are dropped.

Tests included a variety of masses, radii, and periods:

- Masses: 0, 113, 200, 313, 500 g; each plus 27 g for the bar and hanger
- Radii: 75, 100, 125, 150 mm
- Periods: 2, 3, 4, 6, 8 seconds

A. Deflection Correction

An expected characteristic of a proportional controller is that deflection from target position increases with load.

We will implicitly model gear friction by considering lifting and lowering motions separately. The "required torque" parameter used here only accounts for the gravitational and inertial forces, but not frictional effects.

$$defl_{up} = -0.138 \cdot torque + 0.0713 \cdot speed$$
$$defl_{down} = -0.0520 \cdot torque + 0.0777 \cdot speed$$

The $defl_{up}$ equation specifies deflection when moving upward, in opposition to gravity, where the servo must over-

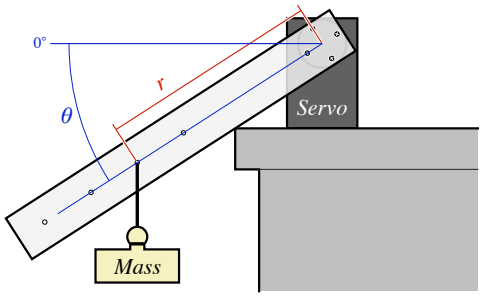

Fig. 8: Torque testing rig, servo is clamped to the table.

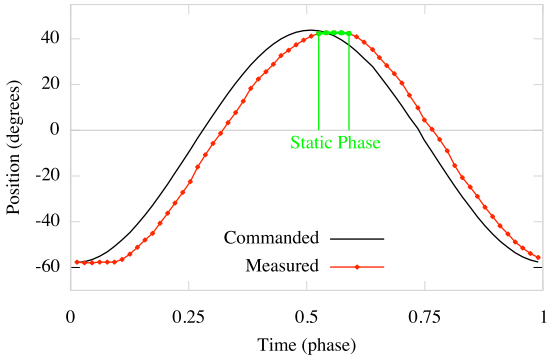

Fig. 9: Example measurement indicating position over time, highlighting the plateau where static gear friction takes hold.

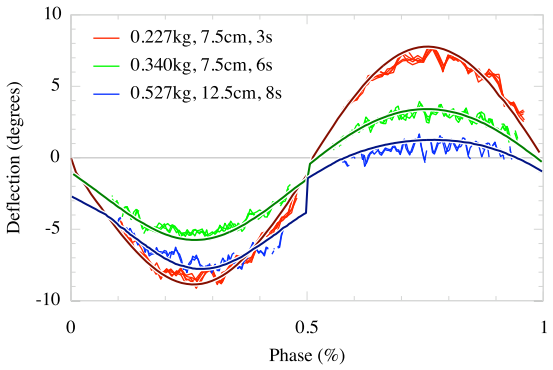

Fig. 10: Prediction of deflection (position error) while lifting and lowering weights. Black is predicted, colored lines are sampled.

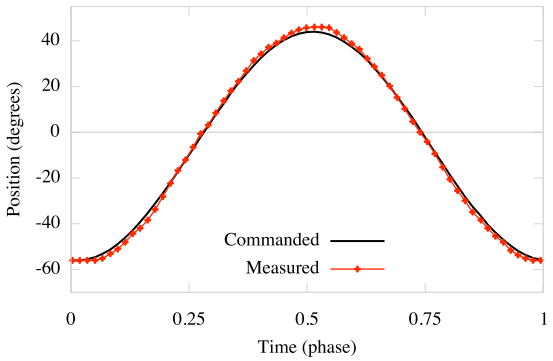

Fig. 11: Application of the deflection prediction to the servo commands yields more accurate motion

come both the external force (gravity and inertia) and gear friction. $defl_{down}$ specifies deflection when moving downward, where gear friction opposes the external force and the servo does not need to do as much work. The speed parameter in both cases is similar, and indicates latency of the servo response. (If moving at constant speed with no load, the servo would still lag behind the target position.)

This deflection value can be interpreted either as a preemptive error correction measure (an offset applied to commands to yield desired position, given torque and speed) or as a torque control strategy (an offset from current position in order to produce a desired torque and speed).

B. "Free Spin" as Torque Control

Another way to emulate torque control is the "free spin" mode, as demonstrated in Fig. 12.

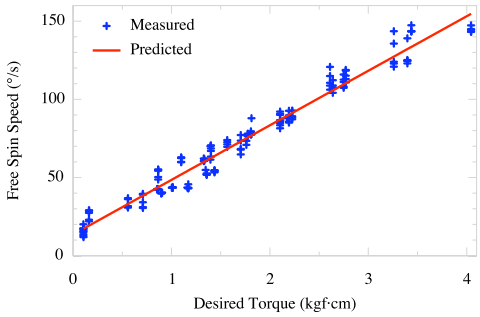

Fig. 12: Free Spin Speed as torque control

To produce this data, the servo "speed" parameter was slowly increased until it was able to lift a weight from rest. The speeds shown here are after applying the generic speed calibration from the earlier section. The conversion from desired kilograms-force-centimeters to degrees per second command is determined to be a linear factor of approximately 34.9, with some minor per-servo constant offset (here, 13.6°/s) to counter gear friction from using the aggregate speed calibration.

An unfortunate restriction of this type of control is that the speed control parameter is limited to ±150°/s, which maps to a maximum torque of 4.2 kgf·cm, although the servo is capable of significantly greater torque. The servo will also cap its torque if the associated rotational speed limit is reached.

C. Load Feedback Calibration

It would be valuable to use the servo load feedback to measure external loads so that their dynamics can be modeled. For example, the deflection prediction described above depends on knowing the applied torque. So if a robot arm is to lift an object accurately, the mass and position of the object is needed to compute its inertia and gravitational force. One way to do this is to lift an object against gravity, and analyze the feedback to determine the external influence.

Dynamixel servos provide load feedback based on the servo's duty cycle. This gives an indication of how much torque the servo is applying to the gears, but is not a direct measurement of resulting torque at the axle. The load feedback follows the same ~8Hz update as the speed feedback. Unfortunately, this coarse temporal resolution is compounded by significant noise, making load estimates during fast motions untenable.

Instead we can invert the deflection prediction obtained previously to obtain a torque estimate based on the recorded deflection. Usually we will also know the location of the object, so we can directly solve for the object's mass:

$$\begin{aligned} defl &= a \cdot torque + b \cdot speed \\ \frac{1}{a} \cdot defl - \frac{b}{a} \cdot speed &= torque \\ &= mass \cdot r \cdot g \cdot \cos\theta + mass \cdot r^2 \cdot \ddot{\theta} \\ &= mass \cdot (r \cdot g \cdot \cos\theta + r^2 \cdot \ddot{\theta}) \\ mass &= \frac{a' \cdot defl - b' \cdot speed}{r \cdot g \cdot \cos\theta + r^2 \cdot \ddot{\theta}} \end{aligned}$$

The load feedback from the servo does still provide some additional information, so using both the deflection and the load feedback in the model helps reduce error.

$$mass = \frac{-5.24 \cdot defl + 0.346 \cdot speed - 0.245 \cdot load}{r \cdot g \cdot \cos\theta + r^2 \cdot \ddot{\theta}}$$

The instantaneous estimates over several representative trials is shown in Fig. 13. Although separate models are used for each of the raising and lowering phases to account for opposing frictional effects, the mass estimates are clumped together when lowering the mass. Because gravity does most of the work in the lowering phase, the servo feedback does not provide much information on its workload.

This means the estimates are only useful in the lifting phase, although even then still noisy. However, averaging the estimates over the course of a motion gives an accurate estimate of the mass. (Fig. 14)

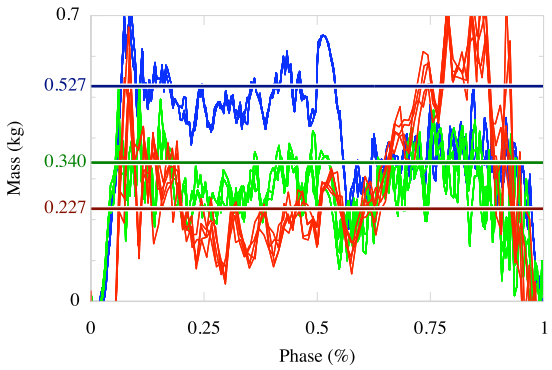

Fig. 13: Recovery of attached mass given known radius of revolution. Black lines are the ground truth, colored lines are instantaneous estimates over several runs.

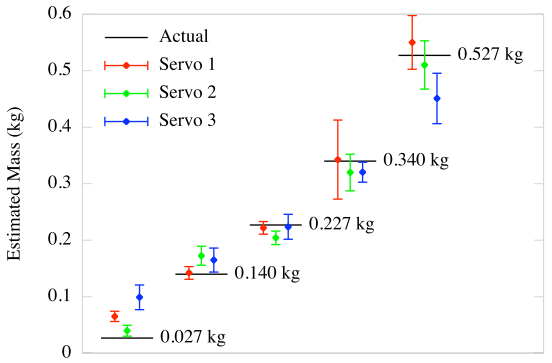

Fig. 14: Std. deviations of mass estimates for three servos sharing a global fit, each point encompasses 20 permutations of speed and radius.

VII. Identified Issues

With a 1 megabaud communication line, the servos are capable of sampling position feedback at very high temporal resolution. Many USB-to-serial chips (such as FTDI's FT232RL) are configured to buffer communication data received from the serial line up to 16 milliseconds, although data sent to the serial line is unbuffered. This configuration limits servo polling to 62.5 Hz. The tests shown here utilized a single servo at a time, and were polled at 31.25 Hz.

On the other hand, a hexapod robot with 18 servos would only have a polling frequency of 3.5 Hz. One improvement is to distribute several read requests over the buffer period, and then read all of the responses en masse when the buffer is flushed. We have been able to reliably poll up to three servos per flush with this method, thus tripling the poll rate when using multiple servos.

Regarding maximum torque, AX-12 servos are able to hold significantly more than their rated 12 kgf·cm when powered at 12 volts. However, they stall significantly lower, around 8-9 kgf·cm. This means that although a large applied load can be resisted from a given position, if the servo is moved, it may be unable to later reclaim its original position.

VIII. Future Work

Different voltage levels have been noted to change the torque characteristics of these (and other) servos. In practical terms, this means a battery powered robot will perform differently as its battery runs down. Since all tests were performed with a constant power supply, this is unquantified. Similarly, the role of temperature on the motor efficiency has not been examined.

The servo controller parameters have been left at their default values during these tests, but adjusting these parameters may allow more accurate feedback in some situations. For instance, when attempting to measure external load, decreasing the proportional response should allow the servo to deflect further from the target position. This should result in more precision for the corresponding load estimate.

Finally, there is a newly introduced wCK servo series found in the RoboBuilder kits, which offer features similar to the Dynamixel. These were not yet available when this work initiated, but a comparison of the performance of these servos would be valuable.

IX. Conclusion

A methodology for modeling and calibrating key parameters of fully digital servos has been described and demonstrated on a specific model. The servo can be used in each of position, speed, and torque control modes, with calibrated feedback for closed-loop interaction.

The result of this work is that these servos may be better utilized in a variety of robotics applications, fostering better collaboration and reproducibility of results by the use of common, off-the-shelf parts.

References

[1] http://www.openservo.com/

[2] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset: Design of a modular snake robot, Intelligent Robots and Systems, 2007

[3] http://www.hitecrobotics.com/manager/control/down_file.php?upFile=HITEC%20HMI%20Robot%20Servo%20General%20informationV1.00_1.pdf&tableName=board_14

[4] F. Lange and G. Hirzinger. Learning Force Control with Position Controlled Robots. In International Conference on Robotics and Automation, Minneapolis, April 1996

[5] C. H. An, C. G. Atkeson, and J. M. Hollerbach. *Model-Based Control of a Robot Manipulator.* The MIT Press, Cambridge, MA, 1998

[6] David S. Touretzky, Neil S. Halelamien, Ethan J. Tira-Thompson, Jordan J. Wales, Kei Usui: Dual-coding representations for robot vision programming in Tekkotsu, Autonomous Robots, Vol. 22, No. 4, pp. 425-435, 2007