State Estimation of a Wild Robot Toward Validation of Rigid Body Simulation

James R. Taylor[1] and Evan Drumwright[2]

***Abstract*— There exist few objective measures to evaluate or compare multi-rigid body dynamics simulators. This absence creates uncertainty in simulation capabilities and accuracy. Simulation science has used theory and other simulations (*verification*) and real-world data (*validation*) to evaluate simulation correctness. With respect to rigid body dynamics, ballistic rigid body motion has been *validated*, but simulations involving contact and friction frequently seem to produce results that appear inconsistent with real-world observations; accurate validation has been seldom performed for contacting "rigid" bodies, likely because the observation problem is so challenging (compared to, e.g., fluid dynamics). This paper concentrates on a simple validation scenario for multi-rigid body dynamics with contact and friction, which are essential for simulating robotic locomotion and manipulation. We study collection and estimation of motion data from a mechanically simple but highly dynamic, real-world robot whose motion is primarily driven by contact and friction.**

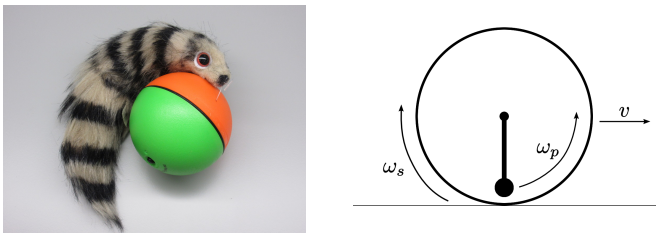
I. INTRODUCTION

Simulation supports design, prototyping, testing, and evolutionary stages of development long before production of any robot; however, confidence among the robotics community in multi-body dynamics simulation remains low. While dynamics simulation of ballistic rigid body motion is well known to be generally consistent with natural behavior, simulations involving contact and friction are prone to producing physically implausible results. Given that the contact and friction models have been validated by mathematicians, physicists, and engineers on simple systems over centuries of study, one must wonder where the software that implements these models fails. Indeed, the past decade has seen the continual release of new simulation software libraries (including one by the second author), each of which promises to eliminate the artifacts but in hindsight has improved the status quo little, if any.

In the past, multi-rigid body simulators have been judged based on performance against a prescribed set of scenarios (see, *e.g.*, [3]) with each scenario designed to evaluate potential modeling errors. These scenarios are a good start but are likely not sufficient; for example, knowledge of multi-body dynamics simulation internals tells us that determining the degree of slope that a box on a ramp begins to slide is unlikely to point to the cause of robotic grasping artifacts.

We believe that those efforts that test elementary aspects of the physical models do remain useful. Nevertheless, this paper focuses on using *validation* (comparison against real world data) rather than *verification* (checking that the software implementation matches theory). Toward that effort, this paper studies and captures data from a *wild*, real-world robot. We provide inertial and visual models of the robot, a guide to capturing motion data from the robot, an initial data set, and state estimates of the real robot. To accomplish this goal, we developed a motion capture process relying on a VICON motion capture system, an inertial model of the robot using SOLIDWORKS, and a visualization of the motion capture data, and we carried out and published data from a set of motion capture sessions.

> Videos rendered using state of the art simulation software (GAZEBO using ODE) will show that this simple system, which should be readily modelable with multi-rigid body dynamics, displays far different behavior from that observed in simulation.



(a) (b)

Fig. 1: (a) An off-the-shelf WEAZELBALL with the two hemispheres, equatorial o-ring, and polar axis switch visible. (b) "Forward" velocity v develops from friction between shell and environment due to angular velocity ω_s, a response to the torque generating pendulum angular velocity ω_p.

The subject is a spherical robot called a WEAZELBALL (WB), shown in Figure 1(a), which meets a variety of desirable criteria: wide availability, low cost, simple mechanism, and complex (wild) behavior. It has been the subject in a number of research efforts focused on controlling aggregate behavior of "wild bodies" [1] [2] [4]. The WB basic principle and design was patented more than a century ago [5], operates by a combination counter-torque drive and offset center-of-mass, and is both underactuated and nonholonomic. A motor drives a pendulum around an axis fixed inside a spherical shell which produces torque in the pendulum and a response torque in the shell. Figure 1(b) illustrates the relationship between pendulum angular velocity, shell angular velocity, and shell "forward" velocity. In an unconstrained environment, counter-torque drives the robot forward with respect to the rotational motion of the pendulum due to friction between the shell and the environment, and the robot

[1] George Washington University, Washington, DC 20052, USA
[2] Toyota Research Institute, Palo Alto, CA 94306, USA

wobbles from side-to-side due to rotating offset center-of-mass. Upon collision with an obstacle, a complex range of motions emerge that are reminiscent of pirouetting, rocking, and tumbling.

We present a process for capturing WB behavior with the desire that the process and data archive might be used, studied, and improved. We believe that successful simulation of real-world WB behaviors will lead to better robotic simulation, because we have found little qualitative agreement between observed WB behavior in simulation and in situ, *even though the* WB *seems well modeled by a multi-rigid body system*. We search for ways to estimate and validate the data collected to build a solid foundation, but we encourage others to recreate and refine this work.

Due to the wild behavior of the WB, we do not expect simulations to be able to reproduce motion capture trajectories, though we do believe it is possible to capture WB behaviors qualitatively within simulation. Rather than comparing telemetry data, WB simulation may instead be validated by comparisons such as whether the characteristic wobble is evinced and whether the exploration is ergodic. This paper does not explore potential comparison metrics; we only attempt to provide a basis for future qualitative comparison.

This paper is structured as follows: instruments used to measure WB state are described in Section II, our process to capture WB state is described in Section III, and data post-processing and analysis is described in Section IV. We provide concluding thoughts in Section V.

II. INSTRUMENTS

Simulation validation relies on analyzing state data, so accurate estimates of WB telemetry data is a requirement. An off-the-shelf WB is not designed for state measurement. The shell has few features that can easily be tracked and the opaque shell obscures pendulum state. Modifications to the WB can be expected to alter the dynamic behavior. For example, adding instrumentation changes mass distribution and adding new features to the shell alters surface geometry and friction characteristics, which may be expected to modify contact behavior. We must accept that some modifications are necessary to measure WB shell and pendulum state, so we select modifications to minimize changes to mass distribution and surface features and to maximize reproducibility in the data gathering process. This section describes a number of instruments used to measure the state of the shell and the state of the pendulum and to increase the accuracy of the measurement process.

A. *Modeling the* WB

The shell and pendulum are the two primary WB components and consist of a number of smaller parts. The shell consists of two hollow hemispheres and an o-ring. The hemispheres thread together to form the shell and the o-ring sits in a groove along the seam formed by the hemispheres. The pendulum consists of plastic housing, AA battery, metal weight, motor, plastic gearing, switch, small screws, wire, and DC motor. We call the WB pendulum the *motor assembly*.

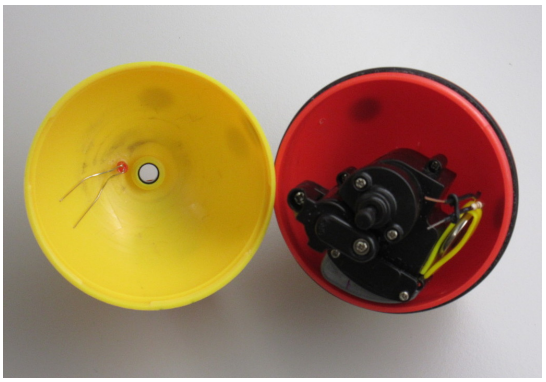


Fig. 2: The interior view of the "free" hemisphere (left-yellow) and the "fixed" hemisphere (right-red). The signal circuit components are also visible: the LED is mounted into the free hemisphere, and battery and leads are mounted to the motor assembly.

Unthreading the shell separates the WB into two pieces: one hemisphere joined to the motor assembly and a hollow hemisphere. We use the terms *fixed hemisphere* to refer to the hemisphere joined to the motor assembly and *free hemisphere* to refer to the hollow hemisphere. Figure 2 shows the separated WB. A metal pin cast into the pole on the inside of the fixed hemisphere acts as mount for the motor assembly. A port cast into the pole of the free hemisphere accepts the power button on the motor assembly which enables the switch to be toggled without opening the WB. The motor assembly rotates around the polar axis when the switch is closed.

We kinematically model the WB as two links constrained together by a continuous revolute joint. We inertially model the shell as a hollow sphere with a diameter of 82mm, a thickness of 3mm, a mass of 46g, and a center of mass located at the center of the shell. Because the motor assembly consists of a number of components with odd shapes and significantly varying densities, modeling the motor assembly as a primitive with uniform density is insufficient. We model the motor assembly using SOLIDWORKS instead. SOLIDWORKS allows parts with different material densities to be modeled individually and then combined into a single assembly. SOLIDWORKS also allows an assembly to be queried for a single inertial tensor and center of mass with respect to a user defined reference frame. Rather than modeling every part in the motor assembly, we simplify by categorizing parts into three groups of solid components: battery, weight, and housing. We measure battery and weight and model them individually. We measure housing, motor, and all other parts as a single component with homogeneous density. We assemble the components and query SOLIDWORKS for the inertial tensor and center of mass using the center of rotation as the center of the inertial frame. We generate Collada files (DAE format) for visualization. We provide all modeling data generated through SOLIDWORKS in the repository.

B. Measuring shell state

We use a VICON array to track the WB as it moves, which allows us to capture the pose of the WB shell over time. We set up a VICON array consisting of sixteen motion capture cameras focused into a cube of space two feet above the ground and roughly six feet on a side. Figure 3 illustrates the camera arrangement we use to support this work. In the center of this space, we marked a large, orthogonal *central axis* on the floor. Around the central axis, we arranged cameras into three rings: a high level ring with cameras focused downward, a mid level ring with cameras focused level to the floor, and a low level ring with cameras focused upward. We include six cameras in the high ring, four cameras in the mid ring, and six cameras in the low ring.

We selected our camera arrangement to share one camera calibration between two activities: *model registration* and *motion capture*. We define model registration to encompass all tasks involved with developing a VICON *tracking model*, discussed in the following paragraph, and we define motion capture to encompass all tasks involved with gathering motion data from a subject. Calibrated cameras may be deactivated and reactivated into the active camera set without the need to recalibrate the array (as long as the cameras are not moved).

A VICON "tracking model" consists of the set of markers registered with the VICON system for a given subject and is used by VICON to derive a subject's pose from the pattern of markers detected by the array. The reference frame for the tracking model, *i.e.* the *model frame*, is defined by a combination of the position of a manually assigned "center" point and the array's origin orientation at the time of model registration. A tracking model must be defined by a unique arrangement of markers in order for VICON's TRACKER software to be able to accurately localize the subject. If two pairs of markers have the same approximate Euclidean distance, VICON will localize the tracking model with an arbitrary pose relative to one of the marker pairs. During registration, if VICON detects a marker arrangement where more than two markers have the same approximate distance, the system will warn of a *symmetric* tracking model. A minimum of three markers must be visible to the array when a sample is recorded; otherwise, VICON either will record a pose with an arbitrary orientation and a position centered with respect to that orientation or will fail to record a pose.

We activate all cameras when registering the WB tracking model, and we activate the high and mid cameras during motion capture. We call the cameras used for model registration the *registration array* and the associated reference frame the *registration frame* and the cameras used for motion capture the *capture array* and the associated reference frame the *capture frame*. The high and low cameras allow VICON to detect markers on the WB from above and below without the need to manipulate the WB during model registration. The low cameras are deactivated in the capture array because they are occluded by the environment.

A standard, spherical motion tracking marker is unsuitable for the WB as placing spheres, even small ones, on the surface of the WB vastly alters its locomotion. We use flat retro-reflective tape instead. Tape does modify contact parameters but the impact is much less than would be the case with spherical markers. We observed that the tape marked WB behaves similar to an unmarked WB.

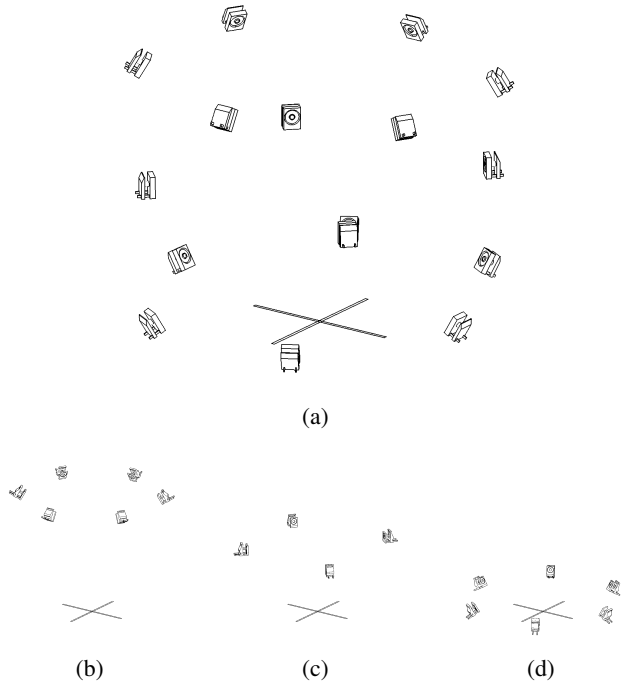


Fig. 3: (a) VICON cameras are centered around the central axes in three rings, are focused on a point approximately 4ft from the floor, and are calibrated together. (b) Six cameras form the high ring. (c) Four cameras form the middle ring. (d) Six cameras form the low ring.

Marker visibility is predominantly subject to occlusion, *i.e.* where a marker is occluded from a camera by an opaque object. Marker triangluation accuracy increases with the number of cameras detecting the marker. A VICON array works best with a marker detected by at least three cameras but will still attempt to "triangulate" a marker detected by two cameras. If fewer than two markers are detected by the array, VICON will not record state.

In lieu of spherical markers, we cut retro-reflective tape into circular markers 12mm in diameter and applied them to the shell. We attempted to minimize the markers' impact on contact properties by using small markers in small numbers. When a flat marker is applied to the shell, the marker forms a dome. A dome marker is less visible than a spherical marker, and a given dome is very often occluded to most cameras. To compensate for reduced marker visibility, we posed cameras to maximize overlap between fields-of-view but we also distributed cameras to maximize viewing angles. Given a sufficient number and distribution of cameras and markers, we found that the array can maintain continuous localization of the shell with acceptable error.

When we tested our initial motion capture process, we

discovered a significant amount of error from the motion tracking model. Our initial array included top down perspective cameras only, which required rotating the WB to add new markers to the tracking model. Using this approach, we were only able to add five markers before VICON detected a symmetric tracking model. With such few markers and using only top down perspectives, we found (1) that VICON recorded a large number of samples where the orientation of the WB was arbitrarily rotated about two markers and (2) that VICON frequently lost track of the WB entirely due to too few markers being visible. We also found that grasping the WB during registration did not allow us to reliably determine the transform from the registration frame to the model frame; this problem made it extremely difficult to objectively determine the pose of the WB from VICON data. To reduce these sources of error, we added a rigid stand, detailed in Section II-D, to the registration process, and we added the mid- and low-height camera rings.

C. Measuring motor assembly state

WB behavior is a product of counter-torque generated by the motor assembly, the rotating offset center-of-mass, and contact/impact between shell and environment. The configuration of the hidden motor assembly drives much of the WB behavior. We considered a number of methods to capture motor assembly configuration, including adding an IMU, inserting an encoder, and casting a transparent shell; however, we expected each of these alterations to affect dynamic behavior by significantly changing inertia and friction.

We choose to capture motor assembly state by adding a simple, lightweight circuit to the WB that generates a visible light signal each time the motor assembly completes a cycle. We add a high definition (HD) camera to our experimental setup to record *signal events* (LED flashes) generated by the circuit. The signal circuit is an independent circuit powered by a 3V button-cell battery. The circuit consists of the button-cell battery, a red LED, a resistor, and a small length of copper wire. We attach the button cell to the motor assembly and rig the circuit so that leads from the button cell contact the LED leads each motor assembly cycle at a constant motor position. The circuit increases the mass of the motor assembly by 5.9% which we consider a minimal change to the motor assembly inertia.

We mount the LED by boring a small hole through the free hemisphere at a dimple located near the power button port. We found the light emitted by the LED is directly observable when the shell is viewed from almost any angle where the switch is visible and is indirectly observable due to LED light reflecting from surrounding surfaces.

During each motion capture session, we record video of the WB using the HD camera at 30 frames per second to capture LED signals. We then post-process our data to synchronize each video with VICON pose data by manually examining each frame for a signal event, recording the frame number of such a signal event, and classifying the signal event.

D. Stand

Introducing a stand, *i.e.* a modified camera tripod, to suspend and fix the WB during model registration facilitates finding a non-symmetric tracking model through using an increased number of markers. Suspending the WB during registration allows us to add the low-height camera ring which improves the likelihood of establishing a tracking model that is detectable from any WB orientation. Fixing the WB allows us to determine the transform from model frame to capture frame. Figure 4 illustrates the positioning of the stand in the registration array. These adjustments enable us to significantly reduce localization errors when compared to our initial approach that did not incorporate a stand.

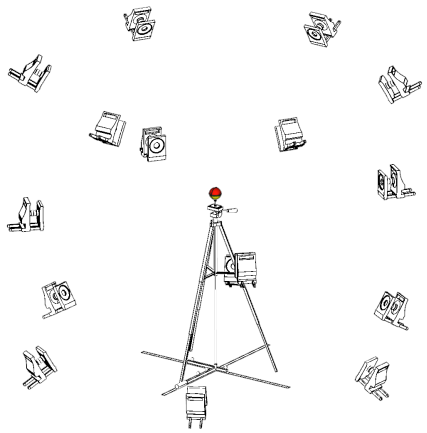


Fig. 4: We activate all three camera rings in the registration array. With the additional cameras and the WB mounted to the stand, we are able to apply markers to any point on the shell without manipulating the WB. We find that marker configurations and tracking models developed using this approach contribute low error.

The VICON reference frame orientation is nominally inherited by the tracking model when that model is registered. We find that if the WB lies in an arbitrary orientation during registration, the transform from registration frame to capture frame is difficult to determine. By mounting and aligning the WB to the stand, we are able to identify this transformation. The stand is aligned to the central axis by centering the stand to the axis origin using a plumb bob and then rotating the stand until one leg is aligned with an axis. Figure 5 illustrates alignment of the stand with the central axis.

We align the WB to the stand using features that we added to the tripod and to the WB. To mount the WB, we attach a clamp with a notch to the tripod head. We align the notch to the axis-aligned stand leg by tilting the tripod head, suspending a plumb bob through the notch, and then panning the tripod head until the notch is plumb with the center of the leg. We level the tripod head and confirm the head is level in both axes. If the head is not level, we adjust the height of the legs until the head is level. We align the WB to the clamp using a pen mark A drawn in an arc from the center of the WB port across the shell equator. We used A both to realign the shell hemispheres whenever we opened the WB for servicing and to align the shell to the stand. When

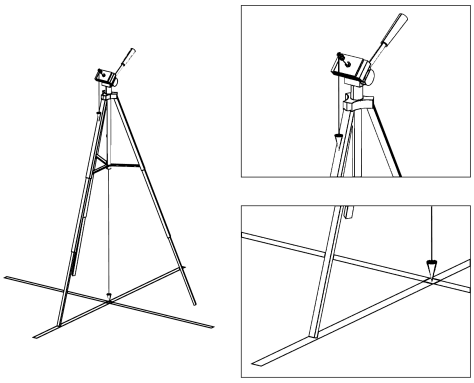
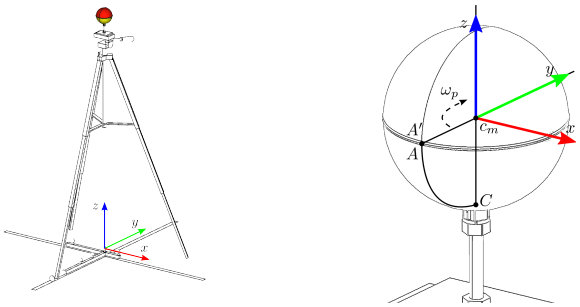


Fig. 5: We align the stand to the central axis by centering the stand to the central axis using a plumb bob suspended from the center of the tripod and rotating the stand until one leg aligns with one axis. We then align the head of the tripod to the axis-aligned leg by panning the tripod head until it is plumb with the leg. Once pan is aligned, we level the head with tilt and leg length adjustments.

mounting the WB, we rotate the WB until A aligns with the notch. Figure 6 illustrates the registration frame origin and the alignment of A to the stand. We choose to mark A to align with the joint angle where the motor assembly closes the signal circuit so that we can assign that joint position as the zero position.



(a) (b)

Fig. 6: (a) We assign the registration frame origin to the central axis and, once the stand is aligned, we mount the WB to the stand. (b) The model frame origin position is manually assigned but the orientation is inherited from the registration frame origin. By aligning mark A to the stand, we are able to transform a fixed point on the shell to any pose recorded by motion capture.

E. Enclosure

We designed a square, inset enclosure (similar in appearance to a table) to limit the area needed for camera coverage and to ensure the WB remains as observable as possible during motion capture. The enclosure is approximately 4ft on a side and 2in "deep". The 4ft surface is large enough for the WB to reach maximum velocity and to exhibit characteristic behaviors, and the enclosure is also small enough that the cameras' fields-of-view overlap (as necessary for motion capture). The enclosure depth is greater than the WB radius, *i.e.* 41mm, to prevent the WB from leaving the enclosure, but the enclosure depth is also shallow enough so that camera placement is not limited to top down perspectives. Due to our marking method, we found that a number of cameras must overlap fields-of-view from many directions. We also found that several cameras must be positioned low enough to track markers close to the enclosure surface, which motivated us to add the mid-height camera ring.

We set up the enclosure with the following steps. First we placed a small, commercially produced table over the central axis as the base. We then placed the enclosure on top of the base. Next, we aligned the enclosure to the central axis. Finally, we levelled the enclosure by inserting shims between base and enclosure until the enclosure was level at the center and in all four quadrants.

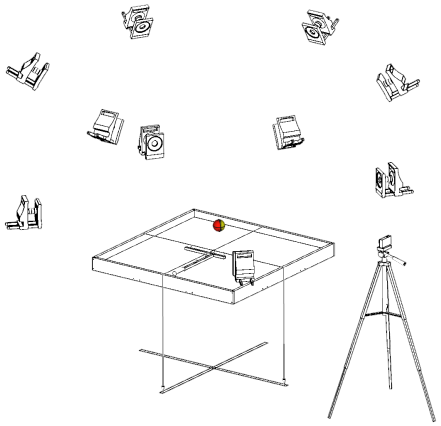


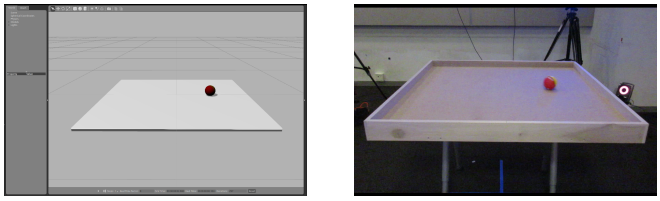
Fig. 7: We activate the high and mid camera rings in the motion capture array. We deactivate the low camera ring because the cameras are occluded by the enclosure. We aligned the enclosure by using plumb lines suspended from the center notches. We centered the wand to the enclosure using mason's line suspended between center notches and then assigned the array origin. We set up a HD video camera with a clear view of the enclosure to record signal events from the motor assembly.

During construction, we cut small notches on top of and at the center of each rail that frames the enclosure. These *center notches* serve two purposes: they enable the enclosure to be aligned with the central axis and they enable the center of the enclosure to be set as the capture frame origin. We aligned the enclosure to the central axis by suspending plumb bobs from the center notches and then adjusted the enclosure's pose until it was plumb with two axes of the central axis. We assigned the capture frame origin to the enclosure center by suspending mason's line between center notches and then aligning the wand with the intersection of the line. Figure 7 illustrates alignment of the enclosure with the central axis and positioning of the wand.

F. Visualization

We have used GAZEBO to visually review streamed VICON state in order to reduce error in data collection and to synchronize between VICON data and HD video data. We used the SOLIDWORKS Collada WB models as a visual model of the WB. We used a GAZEBO plugin to read

VICON state and to update the WB model toward replaying captured data. We have found that replay visualization allows us to assess the quality of VICON camera and WB marker configurations. Through such replay, we found that the configurations of the array and markers have significant impact on the amount of error introduced into the data generated by the motion tracking system. We reviewed HD video with **`melt`** which supports frame-by-frame stepping in the video format produced by the camera. Figure 8 shows a synchronized frame from both visualization and video replay.



(a) (b)

Fig. 8: Two frames from a synchronized visualization show WB state at a given time during motion capture: (a) the frame from GAZEBO visualization and (b) the frame from **`melt`**.

III. MOTION CAPTURE PROCESS

We found that our motion tracking method requires careful planning and testing of both the VICON array and marker arrangement to capture accurate data. After an initial data collection experiment, we noted that casual arrangements of either cameras or tracking markers results in obvious, unpredictable, and difficult to correct errors in the captured data. These errors seem to result from having too few markers visible to too few cameras. To minimize such error, we focused a large number of cameras into the experimental space, varying the camera heights to increase marker detectability, and we used a systematic approach to model registration. We also found that by testing WB motion capture and by visualizing the test data, we were able to tune camera and marker placement to reduce the error in the collected session data.

A. Model registration

Our registration process uses the following steps. We begin by aligning the wand to the central axis and then assigning the array origin frame to the wand. We remove the wand and align the stand to the central axis. We mount and align the WB to the stand. We apply a marker to the shell and register the marker to the current tracking model through TRACKER, which the software then validates. TRACKER may report that the new marker results in a symmetric tracking model. If a symmetric model is detected, we remove the marker, delete the current tracking model, and register all markers previously added (we must delete and re-register the tracking model because a previous marker arrangement may now be detected as symmetric, thereby forcing us to remove markers until VICON accepts a valid marker configuration). Once a valid marker configuration has been attained, we repeat the marker addition process. We were able to add a total of eight markers to the WB using the process, and the resulting tracking model was tracked unambiguously for nearly all samples captured.

B. Motion capture

Our motion capture process involves switching from the registration array to the capture array, aligning the enclosure to the central axis, assigning the capture array origin, aiming the HD video camera, and streaming data from the capture array. Before gathering session data, we gather test data and visualize it. We repeat model registration to improve the tracking model when the test data exhibits clear visual artifacts.

We wish to capture as many signal events from the LED circuit as possible. To ensure that the earliest signal events are visible to the HD camera, we place the camera on the side of the enclosure such that the camera faces the enclosure center and we release the WB into the enclosure such that the free hemisphere faces the HD camera.

Before the WB is brought into the enclosure, we cover the WB with an opaque bag to prevent premature detection. VICON streams data whenever a tracking model is detected, and using the bag reduces occurrences of premature detection. We executed session collection using the following procedure: we place the WB into the bag, we start VICON recording, we start recording HD video, we power on the WB, we place the bag into the enclosure, we release the WB from the bag (and remove the bag from the enclosure), and we record data for an alloted time. Once the session time has elapsed, we stop recording both VICON and video. We carried out a total of ten collection sessions to develop initial sample data. Each session lasted approximately two minutes and was recorded using a 100Hz VICON array and a 30Hz HD video camera.

IV. POST-PROCESSING

After collecting session data, we process raw VICON and video data to identify error in the array, to synchronize the data streams, and to classify motor assembly signal events.

A. VICON analysis

During the visualization process, we identified two anomalous behaviors in raw VICON data as sources of error: (1) the visualized WB oscillates between interpenetrating and floating above the enclosure surface and (2) the visualized WB spontaneously jumps from the enclosure surface with a simultaneous snap change in orientation. We focused our analysis on vertical position z which should be equal to the WB radius, ideally $z = 41$mm, and on change in quaternion orientation $\Delta\theta = ||\Delta \boldsymbol{q}||$ between the current frame and the subsequent frame. Figures 9(a) and 9(c) show that there is noise and position error in all measurements and that spontaneous events are reflected in both z and $\Delta\theta$ measurements as prominent spikes.

We attribute spontaneous events to marker occlusion resulting in a loss of sufficient marker visibility to VICON. We

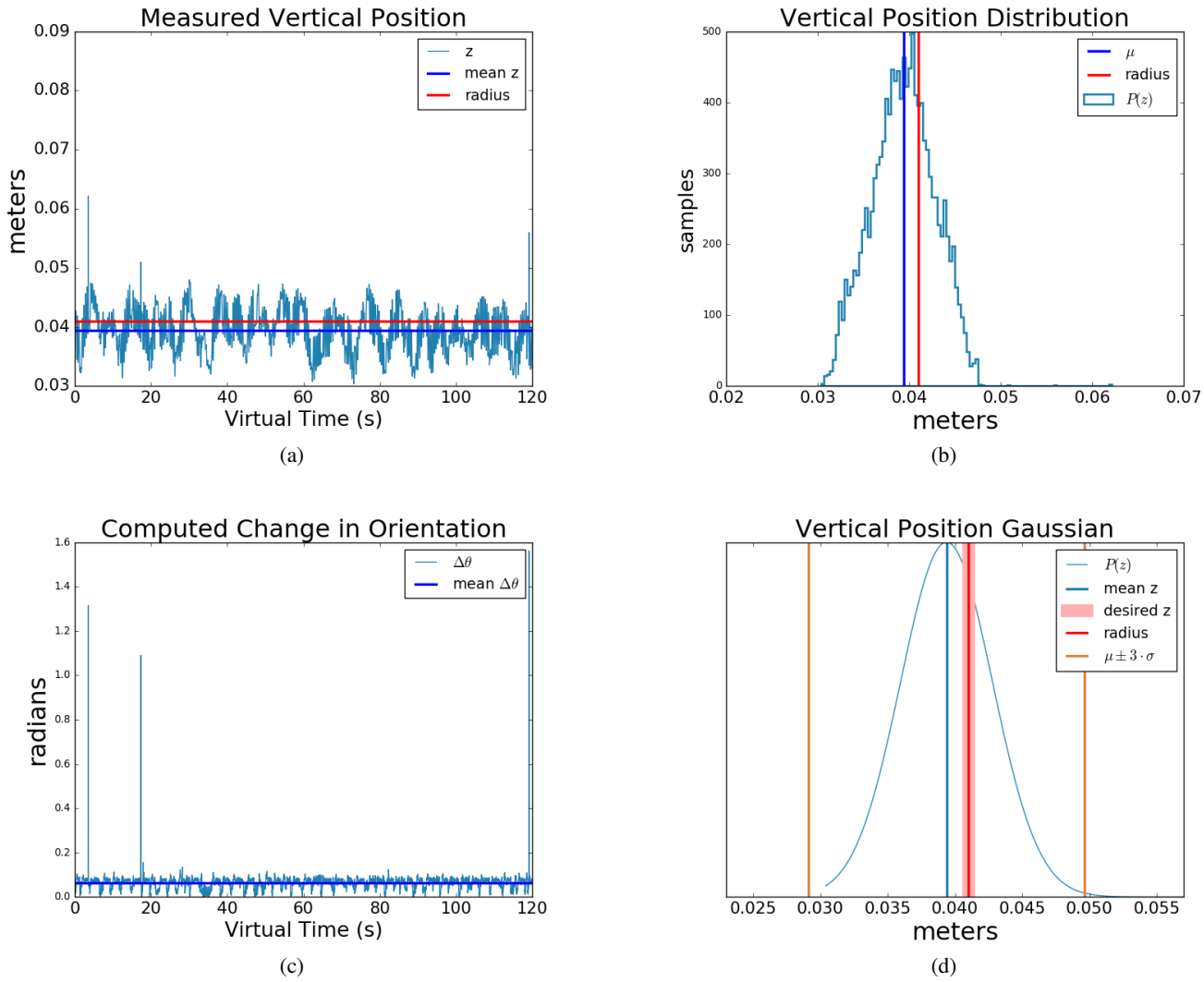


Fig. 9: Analysis of data gathered from the first session produces these plots which are representative of all sessions. (a) Vertical position z should equal WB radius but raw data is noisy, mean should overlay radius but there is a clear difference, and spikes beyond noise suggest the WB jumps above the table at spontaneous samples. (b) Vertical position histogram suggests a Gaussian with a mean left of radius while spikes in (a) appear as right outliers. (c) Change in orientation between each sample exhibits similar error and spikes suggest the WB makes spontaneous snap orientation changes in the same samples as spontaneous jumps. (d) Ideal vertical measurements would be tightly clustered around radius, but the shifted mean, wide bell, and existence of large outliers indicate various error in raw signal data.

found that event incidence is subject to camera and marker placement, so we correlate the number of spontaneous events in a session with the quality of the experimental setup. We found that spontaneous events are infrequent for well-positioned camera arrays and occur primarily when the WB enters a corner. We also found that it is feasible to use interpolation to correct infrequent spontaneous events.

We attribute, at least partially, the oscillation anomaly and the offset between mean and WB radius demonstrated in Figure 9(a) to manual center assignment in the tracking model. We found that manual assignment results in an unknown offset between the tracking model and the subject, which requires estimation of center offset and compensation in VICON data.

The accuracy of our capture array is expressed in Figures 9(b) and 9(d) where the bell shape reflects tracking model and array error, both of which are reduced through fine tuning using test data before session execution. We interpret outliers beyond three standard deviations to the right of the mean as representing spontaneous anomalies. We attribute the difference between the mean measured vertical position and ideal vertical position to error in center assignment, as described in the previous paragraph. These errors could be used during array testing (i.e., before data collection) to tune the capture instrumentation and setup, but we did not pursue that approach in this work.

B. *Synchronization*

Because our process generates both VICON data and video data, it is necessary to synchronize the two independent data sets to produce a single, unified data set. We synchronize by reviewing video data until we identify a collision between

the WB and at least one of the enclosure rails that results in the WB coming to a stop. We then step the motion capture visualization to the same collision. Once the video and visualization are synchronized at this collision, we record the frame number of the video and the time of the capture visualization. We then backtrack to the earliest VICON measurement where we can identify no errors and use that state as the start of virtual time. We compute the virtual time of all VICON and video frames using the start of virtual time and the sampling rate of the VICON system and the frame rate of the video camera. For each motion capture session, we record the following synchronization parameters: the video frame number of the first collision that could be synchronized with VICON data, the VICON time of the same collision in the VICON samples, the total number of frames in the video, the frame rate of the video, the sampling rate of VICON, the start of virtual time in the VICON data, and the video frame that matches the start of virtual time.

We found that differences between the recording rates of the VICON system and the HD camera used in the experiments makes perfect synchronization infeasible. We also found that the video camera sensor type can produce artifacts in the video data. Though our data appears cogent by visual inspection, future capture efforts should anticipate and compensate for these issues.

C. Video analysis

We processed each video stream frame-by-frame to catalog each identifiable signal event recorded by the HD camera. We define an *identifiable signal* as either a direct observation of the lit LED or as an indirect observation of LED light reflected from the enclosure. We found a number of signal events were not identifiable due to occlusion of the LED to the HD video camera and the LED being oriented such that LED light is not noticably reflected by the enclosure; however, given an approximate frequency of the motor assembly we infer that a signal should have been observed. We denote such missing signals as a "gap". Gaps may consist of the loss of one or more signals. We chose not to attempt to infer the number of signals lost in gaps because we discovered that motor assembly angular velocity varies with WB forward velocity, which would require more sophisticated modeling than our initial, constant-angular-velocity assumption can provide. We also found that signals may span two video frames with one of the two signals appearing weak; such cases suggest that the signal duration is approximately equal to the frame rate of the camera plus some small fraction of the frame rate. Table I summarizes the signals captured and classified for all sessions.

TABLE I: HD video LED signal detection

Session	Identifiable Signals	Indirect Signals	Gaps	Two-Frame Signals	Longest Observation (s)
1	249	73	31	39	8.9
2	242	64	32	29	7.567
3	215	37	38	22	4.433
4	232	46	29	18	7.8
5	225	50	34	20	10.33
6	207	35	27	22	8.367
7	265	68	41	46	6.533
8	271	101	37	54	9.567
9	249	58	33	31	6.2
10	265	82	37	39	9.567

This table summarizes our classification of the signal data gathered during session collection. The *identifiable signals* were the number of discrete signals identified on video. *Indirect signals* were those identified through specular reflection. *Gaps* quantifies the number of times that at least one signal should have been observed but no signal was detected. *Two-frame signals* refers to the times that a single signal spanned consecutive video frames. The *longest observation* was the longest period of time where signals were continuously observed.

V. CONCLUSION

We presented a process for collecting data from a wild, robotic system and provide this data in a repository at `https://www.github.com/PositronicsLab/wild-robot`. The repository includes all raw VICON and video data collected, the SOLIDWORKS models and inertial estimates, GAZEBO controllers developed to validate, synchronize and interpolate, and all signal and synchronization data.

Our motion capture process could be improved in several ways, though evolution of motion capture hardware and processes might organically diminish error in future data collection. Using a similar approach, more signal data could be collected by either adding video cameras, adding a high speed camera, using mirrors, or by some combination of this equipment. Signal event classification and synchronization could likely be automated using image recognition.

We will concentrate future work on qualitative validation, since telemetry data is essentially guaranteed to differ between simulation and in situ, using the collected data.

ACKNOWLEDGMENTS

We would like to acknowledge Jack Shannon for assisting in motion capture, Roxana Leontie for creating the circuit, and Jon Torrey for creating the SOLIDWORKS WB model.

REFERENCES

[1] L. Bobadilla, F. Martinez, E. Gobst, K. Gossman, and S. M. Lavalle. Controlling wild mobile robots using virtual gates and discrete transitions. In *American Control Conference(ACC)*, pages 743–749, Montréal, Canada, June 2012.

[2] L. Bobadilla, O. Sanchez, J. Czarnowski, K. Gossman, and S. M. LaValle. Controlling wild bodies using linear temporal logic. In *Proc. Robotics Science and Systems(RSS) VII*, pages 17–24, Los Angeles, USA, June 2011.

[3] A. Boeing and T. Bräunl. Evaluation of real-time physics simulation systems. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 281–288, New York, NY, USA, 2007. ACM.

[4] D. E. Gierl, L. Bobadilla, O. Sanchez, and S. M. Lavalle. Stochastic modeling, control, and verification of wild bodies. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 549–556, Hong Kong, China, May 2014.

[5] T. Ylikorpi and J. Suomela. Ball-shaped robots. In H. Zhang, editor, *Climbing & Walking Robots, Toward New Applications*, chapter 11, pages 546–567. Itech Education and Publishing, Vienna, Austria, October 2007.