Modular Configuration Design for a Controlled Fall

Thomas William Mather

School of Engineering and Applied Sciences

University of Pennsylvania

Philadelphia PA

Email: mathert "at" seas.upenn.edu

Mark Yim

School of Engineering and Applied Sciences

University of Pennsylvania

Philadelphia PA

Email: yim "at" seas.upenn.edu

Abstract—**Much like a falling cat can reorient itself to land on its feet, a climbing robot should also reorient itself to minimize damage during a fall. This paper presents and analyzes the dynamic motion of a modular robot, called CKbot righting itself during a fall. It presents a mathematical model of the falling system that correlates well with experimental reorientation results about one axis. The model also explains why the the process of flipping around is practical only about the long axis of the robot. For more robust orientation correction, a different configuration of CKbot and a new motion plan is presented that corrects for all forms of posture error.**

I. INTRODUCTION

The falling cat problem is the common name for analyzing the dynamic action in which cats are able to turn in mid-air to land on their feet. Work on this problem includes the mathematical formulation and dynamic analysis of jointly connected rigid bodies[6], [12], [10]. In a recent animal study, [8] Jusufi et al. showed that a falling gecko also executes this mid-fall correction using its tail. In those experimental results, the gecko corrected its upside down orientation in $100ms$ or roughly $5cm$ of falling.

As climbing robots become better at climbing, the need for fall correction will become more important. To motivate this assertion, the recent paper by Jusufi, et al. points out how the gecko, nature's most adept climber, is also nature's most adept faller. An experimental robotic framework to test methods of fall correction that is modular in nature would allow a variety of configurations to be tested. CKbot is one such modular system. In fact, demonstrations of modular robots have climbed stairs, poles, fences, porous materials etc. [13].

The falling cat problem is most often formulated as a zero angular momentum problem. As the cat falls, there are no external forces on it, and its angular momentum cannot change. Most of the theoretical focus on the zero angular momentum system is in the context of orientation correction and controllability for satellite systems [4], [10], [12], [3]. There is also significant work in the path planning analysis on this type of system [4], [10], [9], [6], [3]. In [6], Fernandes et al. discuss the falling cat problem, and use it to convey a basis algorithm approach to solve for a semi-optimal control path that flips a pair of rigid bodies, their falling cat model, into a goal orientation. Controlled attitude while falling has also been formulated as a high degree of freedom (DOF) planning problem, in the context of sport diving [7] [2].

This work starts by presenting a derivation of the system dynamics similar to [5], [12]. In addition, it also presents a controllability derivation. One focus of this work is that it takes into account the holonomic constraints, in the form of joint limits that are natural in a system of jointly connected rigid bodies. These constraints are sometimes overlooked in the control setting. In Rui[10] and Walsh[12], reaction wheels are considered for use in attitude tracking. Reaction wheels simplify the control problem because they do not have joint limit constraints, and they do not change the shape and inertia properties of the robot when they spin. This simplifies the control design for orientation tracking. However, in the context of climbing vehicles, systems are mechanically designed to minimize weight and equalize weight distribution. Massive inertial discs are not usually within the weight budget. The other thrust of this work is managing orientation change while considering the inertial distribution of the ensemble while maintaining motions within the actuator joint limits.

Another contribution of this work is the expansion of feasible tasks for a modular system. CKbot, and modular robotics in general, offer a chance to study dynamic behavior on an adjustable robotic sub-system, which until recently[11] has focused on quasi-static motion in the locomotion domain. In the context of a modular system, new joint configurations are simple to create and evaluate. After these systems are analyzed in depth, configuration results can be considered for a more task specific robot; for example a more streamlined climbing or walking robot.

II. MATHEMATICAL BACKGROUND

The system derivation requires some block matrix manipulation. Here is some of the terminology that is used.

$$\underset{\text{axes}}{\text{hinge}} \quad h_x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad h_y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad h_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}$$

$$\hat{\omega} = (\omega \times) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Figure 1 shows the 3 body model of the system. R_1, R_2, and R_3 are the orientations of the three rigid bodies with respect to the world frame in roll, pitch, yaw form. Between the bodies are the single axis joint rotations, $R_x(\theta_x)$ and

$R_y(\theta_y)$. r_i^o is the distance of ith body from the system center of mass (COM). s_{12} is the length from the COM of body 1 to the hinge axis between bodies 1 and 2.
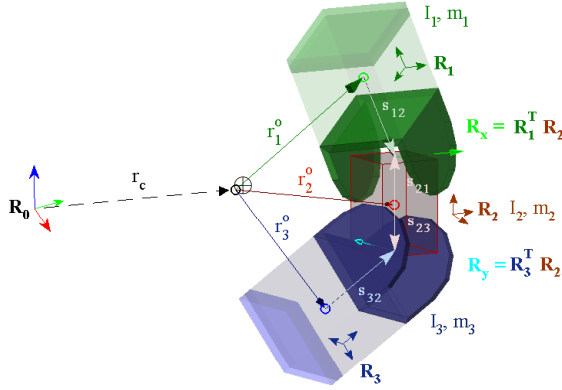
R_0	world frame	r_c	COM of the system
R_i	orientation of body i	r_i^o	distance COM to COM of body i
$R_x,\ R_y$	hinge rotation	s_ij	distance body i to hinge ij
I_i	inertia matrix of body i	m_i	mass of body i

Fig. 1. Three body, two joint model

This system of rigid bodies connected by single or multiple DOF joints can be analyzed with variational methods on the group structure to extract the system of equations[3]. The robot equations of motion have the nonholonomic constraint from the nonintegrability of the angular momentum in the falling system, (neglecting drag). The angular velocities of the rigid bodies are dependent on each other. This leads to the kinodynamic equations of motion. In this case an initial angular momentum of zero is assumed.

$$[R_1\ R_2\ R_3]\begin{bmatrix} \mathcal{J}_M \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = L\mid_{t=0} = [0]^{[9x1]} \quad (1)$$

The elements of $\mathcal{J}_M$ are given by the inertia matrices of each body, I_1, I_2, I_3, and the coupling effects of each body to the others. This $[9 \times 9]$ inertia matrix, $\mathcal{J}_M$, can be determined via the block matrix relationship between internal shape velocities, $v_{1,2,3}^o$, independent of the COM velocity, v_c, and angular velocities, $\omega_{1,2,3}$. The block matrix equation of constraints is,

$$\begin{bmatrix} m_1 I & m_2 I & m_3 I \\ I & -I & 0 \\ 0 & I & -I \end{bmatrix}\begin{bmatrix} r_1^o \\ r_2^o \\ r_3^o \end{bmatrix} = \begin{bmatrix} 0 \\ -R_1 s_{12} + R_2 s_{21} \\ -R_2 s_{21} + R_3 s_{32} \end{bmatrix} \quad (2)$$

The top row of the constraint matrix constrains the COM of the system to be located at the origin. With no external forces on the system, the COM is fixed. The lower rows are kinematic loops around the joints. For example, to get from the COM of the ensemble to the hinge axis between bodies 1 and 2, there are two paths, $r_1^o + R_1 s_{12} = r_2^o + R_2 s_{21}$.

Taking the time derivative of the constraint equation 2 and then swapping the order of the cross products, $R_1 \hat{\omega}_1 s_{12} = -R_1 \hat{s}_{12} \omega_1$) gives the relationship between internal shape

velocities, and angular velocities.

$$\begin{bmatrix} v_1^o \\ v_2^o \\ v_3^o \end{bmatrix} = \begin{bmatrix} m_1 I & m_2 I & m_3 I \\ I & -I & 0 \\ 0 & I & -I \end{bmatrix}^{-1}\begin{bmatrix} 0 & 0 & 0 \\ R_1 \hat{s}_{12} & -R_2 \hat{s}_{21} & 0 \\ 0 & R_2 \hat{s}_{21} & -R_3 \hat{s}_{32} \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

$$= C_{v2\omega}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (3)$$

In Equation 3, $C_{v2\omega}$ is shorthand for the transformation matrix between the internal shape velocities and the angular velocities. To see how these terms contribute to the total inertia, consider the kinetic energy of the system:

$$KE = \tfrac{1}{2}\begin{bmatrix} v_1^o + v_c \\ v_2^o + v_c \\ v_3^o + v_c \end{bmatrix}^T\begin{bmatrix} m_1 I & 0 & 0 \\ 0 & m_2 I & 0 \\ 0 & 0 & m_3 I \end{bmatrix}\begin{bmatrix} v_1^o + v_c \\ v_2^o + v_c \\ v_3^o + v_c \end{bmatrix} \cdots$$
$$\cdots + \tfrac{1}{2}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^T\begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (4)$$

The velocities of each rigid body are the sum of v_c and the shape velocity v_1^o. By substituting in the relationship between the internal shape velocities and the angular velocities in Equation 3, The kinetic energy becomes:

$$KE = \tfrac{1}{2}v_c^T v_c \sum_i m_i + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^T\begin{bmatrix} \mathcal{J}_M \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (5)$$

and the total rotational inertia of the system, $\mathcal{J}_M$, is,

$$\mathcal{J}_M = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} + C_{v2\omega}^T\begin{bmatrix} m_1 I & 0 & 0 \\ 0 & m_2 I & 0 \\ 0 & 0 & m_3 I \end{bmatrix}C_{v2\omega} \quad (6)$$

With $\mathcal{J}_M$ found, Equation 1 can be manipulated to give the equations of motion. Each rotation is constrained by its neighbors. For example if two bodies are separated by an x-axis hinge their angular velocities will be related as follows:

$$R_x = R_1^T R_2 \quad \overset{d/dt}{\Rightarrow} \quad R_x \hat{\omega}_x = R_x \hat{\omega}_2 - \hat{\omega}_1 R_x \quad (7)$$
$$\Rightarrow \omega_1 = R_x(\omega_2 - \omega_x)$$

Substituting this and the similar relation between ω_2 and ω_3 into Equation 2 gives the following for controlling the angular velocity of the central body's orientation ω_2 :

$$[R_1\ R_2\ R_3]\begin{bmatrix} \mathcal{J}_M \end{bmatrix}\begin{bmatrix} R_x(\omega_2 - \omega_x) \\ \omega_2 \\ R_y(\omega_2 - \omega_y) \end{bmatrix} = [0]^{[9x1]} \quad (8)$$

$$J_T \omega_2 = \begin{bmatrix} R_x \\ I \\ R_y \end{bmatrix}^T\begin{bmatrix} \mathcal{J}_M \end{bmatrix}\begin{bmatrix} R_x h_x & 0 \\ 0 & 0 \\ 0 & R_y h_y \end{bmatrix}\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \end{bmatrix}$$

where J_T is the total inertia with respect to the R_2 frame and h_x and h_y are the hinge axis directions. The full equations of motion are:

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ J_T^{-1}\begin{bmatrix} R_x \\ I \\ R_y \end{bmatrix}^T\begin{bmatrix} \mathcal{J}_M \end{bmatrix}\begin{bmatrix} R_x h_x & 0 \\ 0 & 0 \\ 0 & R_y h_y \end{bmatrix} \end{bmatrix}\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \end{bmatrix} \quad (9)$$

$$\dot{R}_2 = R_2 \hat{\omega}_2$$

In the case of high velocity falling, drag becomes a significant contributing force input affecting the angular momentum. This changes the equations of motion significantly. With the small fall durations that are explored in this work, drag has not been accounted for. This analysis is for three bodies, but can easily be extended to n bodies in a chain.

III. LIE BRACKET CONTROL

The falling cat problem is a non-holonomic control problem. The two joint velocities $\dot{\theta}_x, \dot{\theta}_y$ are dependent on the the body frame angular velocities, $\omega_{2|x}$, $\omega_{2|y}$, $\omega_{2|z}$. This is similar to the much covered examples of the unicycle rolling on a plane, where there is a constraint that takes the form that the planar velocities are mathematically dependent.

The non-holonomic constraint on the unicycle constrains the instantaneous velocity to be in the direction of rolling. To produce an effective net velocity with a perpendicular component, a parallel parking-like maneuver is required. Lie Brackets capture this idea of parallel parking[9]. They show the other directions that the system can move by either employing a switching or an oscillating control. This idea is important here because the Lie Brackets suggest oscillatory controls for producing a change in the output state. An amplitude limited sinusoidal input is a more practical control method due joint limits constraints.

Even if the system has more than three shape space inputs for controlling the attitude output, Lie Bracket control would still be preferred, because the joint limits that are present in the system. In an attempt to roll (a direction perpendicular to the hinge axis) a naive gradient descent controller, would cause the configuration to fold in on itself, turning hinges until they ran into their joint limits.

The equations of motion for this two input DOF "falling cat" robot in Figure 1 can be expressed as.

$$\dot{q} = \begin{bmatrix} B_x & B_y \end{bmatrix} \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \end{bmatrix} \qquad (10)$$

$$B_x = \begin{bmatrix} J_T^{-1} \begin{bmatrix} R_x \\ I \\ R_y \end{bmatrix}^T \begin{bmatrix} \mathcal{J}_M \end{bmatrix} \begin{bmatrix} R_x h_x \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} \qquad (11)$$

$$B_y = \begin{bmatrix} J_T^{-1} \begin{bmatrix} R_x \\ I \\ R_y \end{bmatrix}^T \begin{bmatrix} \mathcal{J}_M \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ R_y h_y \end{bmatrix} \end{bmatrix} \qquad (12)$$

The Lie Bracket is expressed as,

$$[B_i, B_j] = \frac{\partial B_i}{\partial q} B_j - \frac{\partial B_j}{\partial q} B_i$$
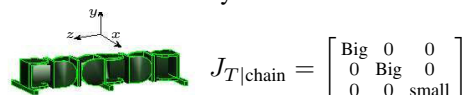
In this case, the bracket simplifies to the following,

$$[B_x, B_y] = \frac{\partial B_x}{\partial \theta_y} - \frac{\partial B_y}{\partial \theta_x}$$

The control vectors, B_x, B_y are only dependent on the shape variables, and the opposite control vector looks like the identity matrix in the shape variables. To prove controllability over this system, it is sufficient to show that the lie brackets, $\{B_x, \ B_y, \ [B_x, B_y], \ [B_x, [B_x, B_y]], \ [B_y, [B_x, B_y]]\}$ span the space to form a five element basis set over the hinge angles and the orientation at all hinge angle values.

The second order bracket is also relatively simple.

$$[B_x, [B_x, B_y]] = \frac{\partial}{\partial \theta_x} \left(\frac{\partial B_x}{\partial \theta_y} - \frac{\partial B_y}{\partial \theta_x} \right)$$

The controllability result is not novel and has been discussed in the literature before. Rui [10] presents a complete discussion of these results. Though controllability is verified and implies that any orientation is achievable, the hidden caveat is that the magnitude of the corresponding second order lie brackets is very small. For a long chain robot, the ability to rotate in pitch and yaw is greatly diminished with the length of the spine. This can be seen as the inertia multiplier J_T^{-1} in Equation 9 greatly reduces the magnitude of movement about the x and y axes.

$$J_{T|\text{chain}} = \begin{bmatrix} \text{Big} & 0 & 0 \\ 0 & \text{Big} & 0 \\ 0 & 0 & \text{small} \end{bmatrix}$$

Fig. 2.   Inertia of an chain robot

The magnitude of the second order brackets, $[B_x, [B_x, B_y]], [B_y, [B_x, B_y]]$ are very low for the long chain configuration. Though this effectively limits practical controllability, there is an interesting consequence of this result. Due to the low sensitivity in getting the two hinged model cat to pitch or yaw, rotational paths in the joint space, $(\theta_x, \ \theta_y)$ produce an ensemble roll motion with very little noise in pitch and yaw. This is interesting because the orientation change for the actual falling cat is simple to execute. Out of sync motions of an animal's head and tail, modeled here as θ_x and θ_y, will roll the animal over.

A. Maximizing Roll Velocity

Fernandes, in [6], includes a 2-axis joint numerical example that is qualitatively equivalent to a roll-only gradient descent control on the same system. For the modular system in the following sections, a fast roll input simplifies to:

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \end{bmatrix} = \left\{ \begin{bmatrix} B_x & B_y \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ u \end{bmatrix} \ \middle| \ \left\| \begin{matrix} \dot{\theta}_x \\ \dot{\theta}_y \end{matrix} \right\|_\infty = \omega_{\text{SERVOmax}} \right\} \qquad (13)$$
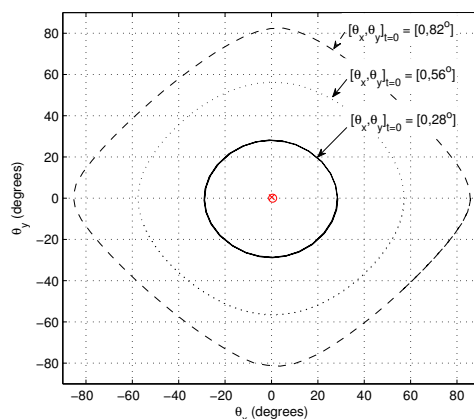

Fig. 3.   Maximumal Roll Velocity Curves using the control in (13) for different intial conditions

By using the transpose of the $[B_x, B_y]$ matrix, this maximizes the $\omega_{2|z}$ response. That is, at every point in the space, the controller chooses a shape velocity to maximize $\omega_{2|z}$. The paths, in Figure 4 generated by the control input in Equation 13, are circular around the origin. This causes the system to roll around the z-axis. Note how the lie bracket $[B_x, B_y]$ suggests this control method for rolling. Figure 4 shows the level sets of the $[B_x, B_y]$ bracket in the $\omega_{2|z}$ direction. The level sets are the same shape as the gradient descent path above. Travelling on the level sets of the Lie Bracket directions is equivalent to the fast roll control.
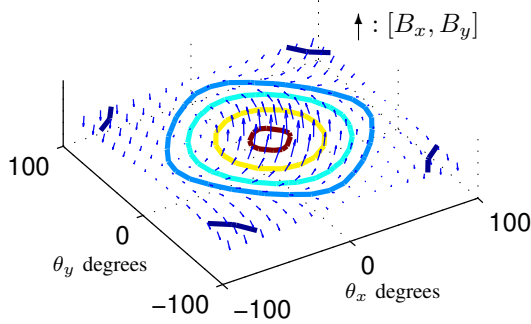

Fig. 4. Level sets of the Lie Bracket $[B_x, B_y]$ in the ω_{2z} direction. The vectors are the Lie Bracket direction in the ω direction. Each bracket vector is calculated at the corresponding location in the joint space.
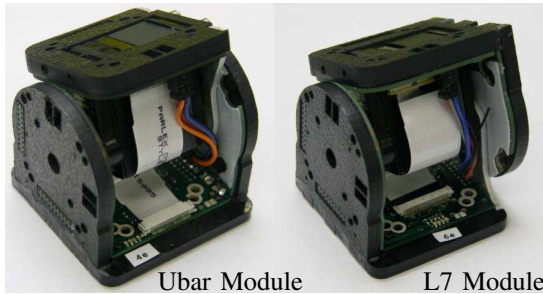
IV. MODULAR ROBOTIC SYSTEM - CKBOT

Fig. 5. Two CKbot Modules with slightly different kinematics. The L7 can have a inline twist the Ubar cannot.

The modular system used in these experiments is CKbot. CKbot modules have a 6cm cube form factor. Figure 5 shows two of the building block modules, the Ubar and the L7. Each has four connecting faces that can be mated with another module via a 20pin header. Each has one 94359 Airtronics servo that gives a maximum 1.0 Nm of torque and has a max speed of $60°/0.1$ second. The module's electronics include a PIC microcontroller that uses CANbus communication to talk with other modules.

The configuration in the experiment is shown schematically in Figure IV. It is a slight modification of an earlier CKbot configuration in which a scalable dynamic centipede hops[11] shown in Figure 6. In the falling experiments, the six module centipede is modified so that its internal joints are orthogonal from each other.

V. EXPERIMENT - DROPPING THE ROBOT

The output response of the system depends on the angular velocity of the joints. Servo motors are geared down and have a proprietary control loop affecting their response. Before any drop tests, the dynamic servo response versus command has to be characterized. This is done by a machine vision system which extracts a model by observing visual fiducials placed on the modules as they oscillated back and forth with an inertial load. This data is compiled to create a one pole Simulink model with feedback and velocity saturation. This model is experimentally verified over a range of inertial relatively small loads, with the servos reaching their maximum velocity within 20ms.

TABLE I
GAIT CONTROL TABLE

$t_{(ms)}$	$\theta_y°$	$\theta_x°$
0	$0°$	$0°$
20	$0°$	$46°$
40	$-69°$	$46°$
100	$-69°$	$-57°$
200	$69°$	$-57°$
340	$69°$	$69°$
480	$-35°$	$69°$
600	$-35°$	$0°$
700	$0°$	$0°$

A *gait table* is used to control the modules. A gait table is a sequence of position commands for each module in the system. Modules process the gait tables by interpolating position commands sent to the servos at the maximum command rate of 60Hz. Fast motions on the gait tables require fewer interpolants. The falling experiment requires the servos to operate at top speed. Therefore the gait table has no interpolants. The gait table in Table I and corresponding path in joint space is plotted in Figure 8. Each dot on the path occurs at 10ms intervals.

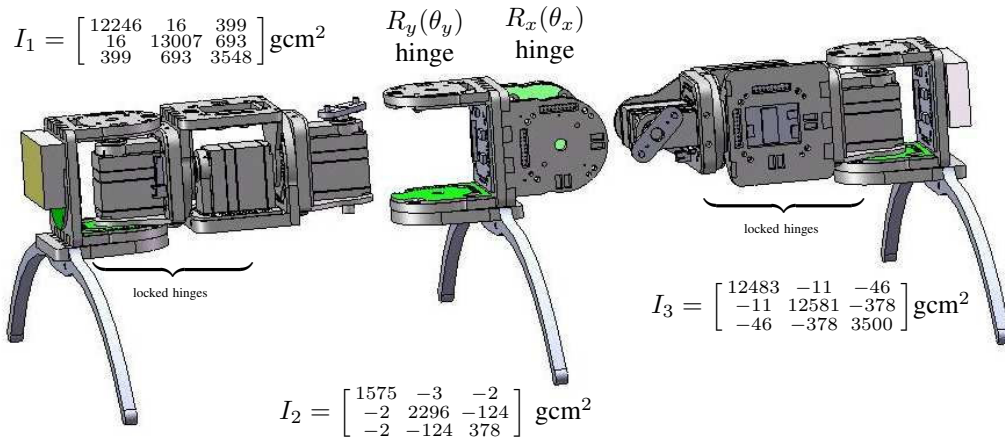Fig. 6. Dynamic centipede with springy legs for jumping[11]

$$I_1 = \begin{bmatrix} 12246 & 16 & 399 \\ 16 & 13007 & 693 \\ 399 & 693 & 3548 \end{bmatrix} \text{gcm}^2 \qquad \begin{matrix} R_y(\theta_y) \\ \text{hinge} \end{matrix} \quad \begin{matrix} R_x(\theta_x) \\ \text{hinge} \end{matrix}$$

locked hinges

locked hinges

$$I_3 = \begin{bmatrix} 12483 & -11 & -46 \\ -11 & 12581 & -378 \\ -46 & -378 & 3500 \end{bmatrix} \text{gcm}^2$$

$$I_2 = \begin{bmatrix} 1575 & -3 & -2 \\ -2 & 2296 & -124 \\ -2 & -124 & 378 \end{bmatrix} \text{gcm}^2$$

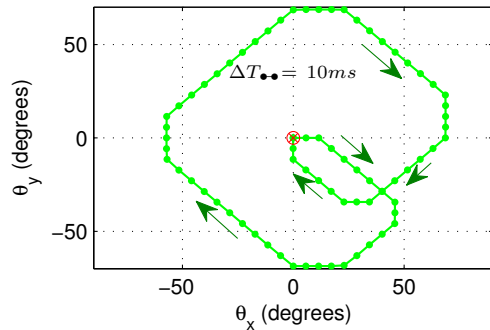Fig. 7. CAD model of experimental CKbot configuration with inertia matrices

Fig. 8. Path in shape space of two hinge joints θ_x, θ_y

A. Experimental Procedure

The modular system is programmed to hold itself static for two seconds while an experimenter prepares to drop it from a fixed height. Lightweight fishing line attaches a handle to a small hole in the central leg structure above the COM. LED's signal the experimenter supporting the system to release. A high speed (240FPS) camera captures the motion of the system as it falls. The drop height is roughly 290cm and falls into 15cm of padding on the floor. This gives a 275cm drop height which corresponds to $\sim$0.75 seconds of drop time and final speed of 7.3m/s

B. Results

As can be seen in in Figure 9, composited stills paired with simulation from one run. It is apparent from the data that flipping over (rolling in our case) as a cat can be achieved and that the mathematical and simulated models match well for this case. A slow motion video of this result is included in the conference proceedings as well as at [1]. Stills of the experiments are shown in Figure 9. Close inspection shows that the device was dropped about 60ms too early.
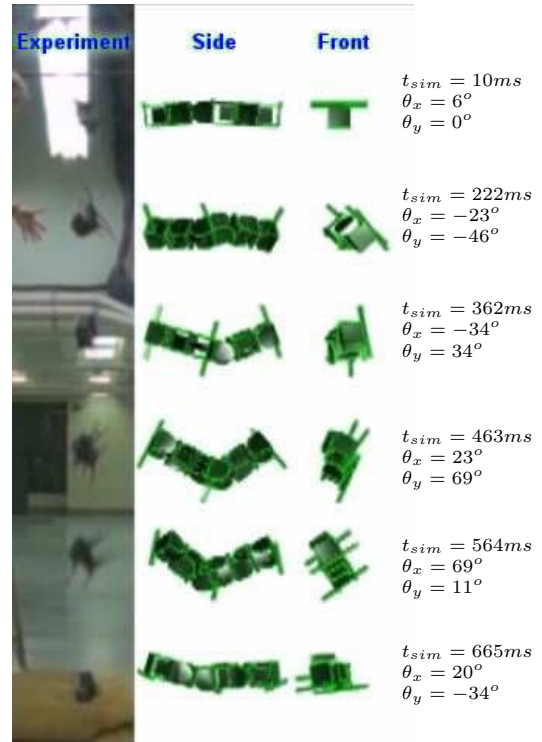

Fig. 9. Experiment snapshots including video (left), simulation right) and simulated measurements.

VI. FALLING ROBOTIC DESIGN

While we have shown reorienting about roll, it would be interesting to reorient about other axis as well. One goal of this work has been to propose a robotic configuration that could most effectively reorient during free fall. In particular, a configuration similar to the CKbot dynamic centipede (Figure 6) which is exploring jumping by using springy legs. These legs would also be able to absorb the energy from a fall.

The Lie Bracket analysis shows that the system is controllable because the set of
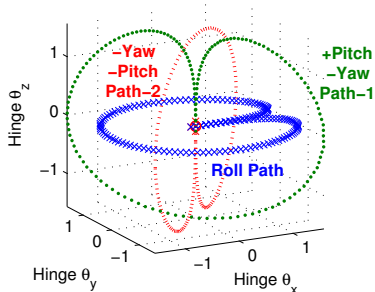
Fig. 10.   Preliminary 3 Joint $R_x R_z R_y$ study

$\{B_x, B_y, [B_x, B_y], [B_x, [B_x, B_y]], [B_y, [B_x, B_y]]\}$ spans the space. However, inspection via theory and simulation of the system shows that the achievable angular velocities using the double bracket type controls are insufficient for directions other than roll.

Under these mathematical constraints, a falling configuration should be constructed with enough DOFs to generate single order lie brackets that span the workspace. As explained in the Section III, first order brackets give the system the freedom to move relatively quickly without running into joint limit constraints.

With this in mind, the simulator is set-up with an augmented modular configuration of an additional R_z hinge. This modular configuration is similar to the experiment configuration with a 3rd active R_z module in the place of the middle body. Intuitively, this creates a ball and socket joint between two off set inertias. This construction shows better maneuverability around the pitch and yaw rotational axes. With the third hinge, the inertial loads of the head and tail can be used to change the components of J_T so that J_T^{-1} does not nullify the response. Figure 10 shows some preliminary results from control paths for the $R_x R_z R_y$ centipede. The roll loop is similar to the loop shown in the experimental results above and rotates the device $180°$ The two other loops create small rotations about the other axes.

VII. CONCLUSION & FUTURE WORK

We have shown that a modular robot system with hinge axis that bend about yaw and pitch in elongated configurations can reorient about roll during a fall. However, a major contribution of this work is the analysis of these elongated systems which shows that while these structures can efficiently spin about the axis parallel with their spine, other rotations are much more difficult. This is due to large inertia about the pitch and yaw axes coupled with a lack of access to the high gain actuation in the shape space due to holonomic constraints. By including more orthogonal degree of freedom joints, the system has greater access to oscillatory controls that adjust the orientation of the robot.

It is important that these joints be centrally located on the robot. The inertia models of the offset bodies are important, but the more significant inertia contribution comes from the mass distance^2 product. If the joints are at the far ends of the

system, they won't be able to move the large inertial load they are attached to. This implies that "tail" type inertias are not potentially useful, however tail type inertia loads do have a significant advantage in that their final orientation is usually unimportant to the system.

Unlike in simulation or experiment, the falling state is not well defined. The assumption that the system begins with zero angular momentum is convenient mathematically, but unlikely. Falling off a wall or a tree limb often implies an initial angular momentum and random orientation. This, along with falling from different heights, articulated legs, and dealing with drag are left to future work.

REFERENCES

[1] Video available at *http://modlabupenn.org/research/dynamic-robots/*, also on *http://www.youtube.com/watch?v=-lEdgu0ZEAY*, 2008.

[2] J.V. Albro, G.A. Sohl, J.E. Bobrow, and F.C. Park. On the computation of optimal high-dives. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 4, pages 3958–3963 vol.4, 2000.

[3] C.K. Chen. *Control of coupled spatial multibody systems with nonholonomic constraints*. PhD thesis, Case Western Reserve University, 1993.

[4] C.K. Chen and N. Sreenath. Control of coupled spatial two-body systems with nonholonomic constraints. In *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 949–954 vol.2, Dec 1993.

[5] I-M. Chen and G. Yang. Automatic model generation for modular reconfigurable robot dynamics. *Journal of Dynamic Systems, Measurement, and Control*, 120(3):346–352, 1998.

[6] C. Fernandes, L. Gurvits, and Zexiang Li. Near-optimal nonholonomic motion planning for a system of coupled rigid bodies. *Automatic Control, IEEE Transactions on*, 39(3):450–463, Mar 1994.

[7] J.-M. Godhavn, A. Balluchi, L. Crawford, and S. Sastry. Path planning for nonholonomic systems with drift. In *American Control Conference, 1997. Proceedings of the 1997*, volume 1, pages 532–536 vol.1, Jun 1997.

[8] A. Jusufi, D.I. Goldman, S. Revzen, and R.J. Full. Active tails enhance arboreal acrobatics in geckos. *Proceedings of the National Academy of Sciences*, 105(11):4215–4219, 2008.

[9] R.M. Murray and S.S. Sastry. Nonholonomic motion planning: steering using sinusoids. *Automatic Control, IEEE Transactions on*, 38(5):700–716, May 1993.

[10] C. Rui, I.V. Kolmanovsky, and N.H. McClamroch. Nonlinear attitude and shape control of spacecraft with articulated appendages and reaction wheels. *Automatic Control, IEEE Transactions on*, 45(8):1455–1469, Aug 2000.

[11] J. Sastra, W. G. Bernal-Heredia, J. Clark, and M. Yim. A biologically-inspired dynamic legged locomotion with a modular reconfigurable robot. In *Proc. of DSCC ASME Dynamic Systems and Control Conference*, Ann Arbor, Michigan, USA, October 2008.

[12] G.C. Walsh and S.S. Sastry. On reorienting linked rigid bodies using internal motions. *Robotics and Automation, IEEE Transactions on*, 11(1):139–146, Feb 1995.

[13] M.H. Yim, S.B. Homans, and K.D Roufas. Climbing with snake-like robots. In *IFAC Workshop on Mobile Robot Technology*, Jejudo, Korea, May 2001.