Flippin' Felines: Controlling a Cat Model to Land on its Feet

Team Members
- Chris Dembia
- Sean Sketch

Contents
- Motivation
- Modeling
- Optimization
- Results & Conclusions
- Next Steps
- Contributions
- Tutorial and Source Code
- References

To learn how to do this project on your own, check out our Tutorial: Studying the Cat-Righting Reflex.

Motivation
Between 1998 and 2001, researchers [3] documented 119 cases of "feline high-rise syndrome", the tendency of cats to fall from heights greater than two stories. This often occurs because the animal, cooped up in an apartment, is distracted by prey outside an open window. For each case, the cat's fall height (in high-rise floors) was recorded and its resulting injuries were scored. The data is displayed in the following plot:

The feature of interest is the RED bar, representing the average injury score for cats falling from 7 stories or higher. This "7th and more" grouping suggests that cat injury does not continue to increase with fall height. This finding has puzzled researchers. The predominant explanation cites the cat's uncanny ability to flip in midair and land on its feet, known as the "cat-righting reflex". Because the animal can perform this maneuver so quickly, it has time before landing to spread its body like a parachute, increasing drag and decreasing its terminal velocity, thereby reducing the force of impact.
Feline high-rise syndrome certainly piques one’s interest about the cat-righting reflex. However, it is the apparent simplicity of the cat-flipping problem that is attractive from a research perspective. A surprising number of papers have investigated the cat-righting reflex. First to tackle the problem were Kane and Scher in 1969 [2]. They proposed a simple mechanical model – namely, two bodies connected by a no-twist joint – possessing “salient features” of the motion of a flipping cat; however, they did not discuss control of the model. Thirty years later, in 1998, Arabyan and Tsai [1] added 7 more bodies (head, neck, legs, and tail) to the model proposed by Kane and Scher and used a two-stage control algorithm to execute the flip. Most recently, Ge and Zhang [4] returned to a 2-segment model, approaching the problem from the perspective of nonholonomic control. Our investigation will take the following approach:

- model the cat and, based on this model, define what it means for the cat to "flip"
- add actuators to each of the model’s degrees of freedom
- based on the "flip" criteria, define an objective function for optimization
- optimize the control of each actuator by minimizing the objective function
- compare the resulting model motion to the actual motion of a flipping cat
- modify the model accordingly and repeat

Following these steps, we aim to answer two questions:

1. When falling upside down, what control strategies does a cat use to flip itself over and land on its feet?
2. What is the minimum model of a cat that will flip in a biologically realistic manner? For example, does the cat need legs to perform its acrobatics? What about a tail? Can it flip without being able to twist its spine?

In addition to these basic research questions, we believe that the cat-flipping problem provides an ideal framework for learning and teaching OpenSim. Our learning is on display throughout this report. Teaching is the focus of the provided Tutorial: Studying the Cat-Righting Reflex.

Modeling

Focusing on the second of the two research questions, we defined a cat-modeling spectrum:

On one end of the spectrum, a rigid body will not flip (from rest) without violating the conservation of angular momentum. On the other end of the spectrum, a real cat most certainly flips. But what about two bodies pinned together? What about more complex models? How accurately do we need to model the animal before achieving a biologically realistic flip?

To systematically explore the spectrum, we defined a “base case” model as two segments welded together (i.e., a rigid body). The base case could yaw, pitch, and roll. To increase the complexity of this base case and advance along the spectrum, we added bodies (legs/tail) and degrees of freedom. We defined the inter-segment degrees of freedom as twist, hunch, and wag.

The legs added to this model could either be fixed relative to their segment or free to pivot (via actuation) about a pin joint. The tail was provided either one or two degrees of freedom. These were equivalent to the inter-segment hunch and wag degrees of freedom; in no case was the tail allowed to twist about its axis.
We wrote a C++ script to automatically generate all model combinations. For each model, torque-based actuators (with spline functions as control inputs) were added to the inter-body degrees of freedom. Coordinate-limit forces (starting at coordinate values extracted from images/video) could be toggled on and off prior to running the model through the optimizer.

Optimization

The primary requirement for a successful optimization is a well-defined objective function. In the cat-flipping example, minimizing the value of this objective should correspond to a biologically realistic flip of the model. To define what it means for a flip to be "biologically realistic", we analyzed high-speed video from a National Geographic special:

From this video, we defined the following objective function: $J = \left( \text{roll} - 180^\circ \right)^2 + \left( \text{twist} \right)^2 + \left( \text{hunch} + 2 \cdot \text{pitch} \right)^2$. The roll term in this equation dictates that the model's front feet are oriented downward. The twist term does the same for the back feet. The combined hunch-pitch term guarantees that the cat lands with a symmetrical arch to its back. During the optimization, this function was evaluated at the end of every forward dynamics integration. If its value was below a preset tolerance, the flip (and therefore optimization) was deemed complete.

Although easy to understand, this objective was not enough for the OpenSim optimizer to efficiently find actuation patterns (i.e., spline points defining the prescribed controllers) that flipped the model, let alone flipped it in a biologically realistic manner. To "help" the optimizer find solutions, we enhanced the objective function by adding terms penalizing the following:

- deviations from biologically realistic hunch, wag, and yaw
- feet unprepared for landing
- "over-twisting" (i.e., twist > 90 degrees)

We provided less weight to deviations from desired velocities and accelerations (typically zero) than to deviations from desired positions (i.e., coordinate values). Moreover, to further enforce the flipped condition, we added "task-space" terms to the objective. These defined vectors for each model segment (anterior and posterior) fixed relative to the segment and originating from the inter-segment joint. For the model to be flipped, this vector needed to achieve a specific configuration in task/operational (as opposed to joint) space. Deviations from the desired configuration were penalized.

Using an XML file to modify the objective-function parameters (e.g., desired coordinate values and weights on these values), we were able to run optimizations on multiple models in quick succession. After many of these runs, we found several models that were able to flip. Each used a slightly different strategy...

Results & Conclusions

Track the YELLOW vectors or legs in each of these looping videos. When they are oriented downward, the cat has flipped. In the upper left is a rigid body (i.e., the "base case") with a two-degree-of-freedom tail. Below that is a model without a tail that can only hunch and wag, not twist; in this case, the rear of the cat acts like a more massive tail. In the bottom left is a combination of the first two models, hunch and wag plus the tail. In the upper right, legs have been added to the model, but the spine has been restricted to only twist. Although it is "getting there", this model does not flip. Finally, the large model has everything except for a tail.

Seeing these results, we came to the conclusion that NO PART OF THE CAT IS "WASTED". Each of the models in the video does flip (or gets close), so the minimum flipping model would be one with two degrees of freedom; say, hunch and wag. Yet, each of these flips has unrealistic characteristics, such as excessive hunching. A full model may be necessary for biological realism, as specified by our second research question.

To answer the first research question, we noted that biologically realistic cat flipping seems to be a combination of two “flipping modes”:

<table>
<thead>
<tr>
<th>Variable-Inertia Mode</th>
<th>Counter-Rotation Mode</th>
</tr>
</thead>
</table>

As its name suggests, the variable-inertia mode relies on changing the bodies’ moment of inertia throughout the fall. In the above clip, the spine can twist. To get one half of the cat to counter-rotate faster than the other, that half’s moment of inertia is reduced by bringing the legs close to the body. The other half’s legs stay normal to the body, keeping its moment of inertia (relatively) high. The counter-rotation mode is for non-twist connections. As seen in the video clip, the tail’s rotation about the axis of the spine allows the body to rotate in the opposite direction. To conserve angular momentum, the less massive (and therefore less inertia) tail rotates significantly faster than the body.

Next Steps

Our work certainly does not represent the final word on the cat-righting reflex. As noted below, our source code can be downloaded from our SimTK project page. Using this as a starting point, we propose the following as viable next steps for the research:

1. Select more accurate modeling parameters (e.g., segment dimensions and mass) based on cat anatomical data.
2. Actuate the model by prescribing kinematics instead of torques. This will allow for more intuitive guessing of initial parameters, thereby accelerating the optimization process.
3. Add contact to the model and get the cat to land stably when falling from an arbitrary (yet realistic) height.
4. Perform inverse kinematics analysis with real flipping cats (or pre-existing videos of real flipping cats). This would provide a true metric for biological realism.

If you are new to OpenSim and just learning the API, perhaps begin with our Tutorial: Studying the Cat-Righting Reflex that teaches how to make models and perform dynamic optimization.
Contributions

Beyond the insights into the cat-righting reflex, we have provided the following tools to the OpenSim community:

1. framework for automatic 'step-wise' model creation
2. framework (based on XML) for rapid configuration of the optimization
3. instructional tutorial covering modeling, coordinate actuators and limit forces, spline-based prescribed controllers, and dynamic optimization

Tutorial and Source Code

To learn how to create the models and perform the optimizations described above, visit our Tutorial: Studying the Cat-Righting Reflex. You can obtain our source code from our SimTK project page.

References