Modeling and Simulating Mechanical Systems on a Transforming Dicycle

By Danaan Metge, BPG Motors

The Uno III is unlike any other vehicle in the world. Originally prototyped as a self-balancing, electric dicycle, the Uno III can transform into a motorcycle by deploying a third wheel on the fly (Figure 1).

![Figure 1. The Uno III dicycle in Uno mode (left) and in motorcycle mode (right). During the transformation to motorcycle mode, the third wheel, stored in Uno mode, is moved forward while the rear traction wheels shift backwards, providing greater stability at high speeds.](image)

Much like aircraft control systems, the Uno’s controllers must manage roll, pitch, and yaw, as well as forward motion and the transformation from dicycle to motorcycle. They must also handle throttle and steering inputs from the rider, as well as side-to-side and front-to-rear shifts in the rider’s position. With five independent motors, six gyroscopes and accelerometers, and four potentiometers onboard, testing and tuning the Uno’s control systems is a complex task, made even more challenging by the need to ensure rider safety at all times.

When we began the Uno III redesign, it was clear that manually tuning controllers and testing on the actual vehicle would be inefficient and risky. Instead, we used Simulink®, Simscape™, and SimMechanics™ to model and simulate the Uno’s mechanical systems.

During real-world tests, things move so fast it is impossible to understand everything that is happening. In simulations, however, we can use Simulink to freeze time and inspect every aspect of the model to get a clear picture of how the mechanics are behaving. We can then see exactly how to improve the control system.

A Brief History of the Uno

Company founder Ben Gulak conceived the idea for the Uno while still a teenager. On a trip to China in 2006, struck by the pollution caused by heavy city traffic, he decided to create an eco-friendly commuter vehicle capable of being driven and stored in congested areas.

Gulak built the original Uno on an angle-iron frame from wheelchair motors, batteries, and gyroscopes. The design won a Grand Award at the Intel International Science and Engineering Fair, and appeared on the cover of Popular Mechanics magazine.
After securing startup funding, Gulak and a team of engineers designed the Uno II, which enabled the relatively low-speed dicycle to convert into a motorcycle for higher-speed operation.

To increase the stability and safety of the Uno, the company embarked on a complete redesign for the Uno III. This redesign included improvements to the transforming technology and the gyroscopic tilt system.

**Modeling the Tilt System**

Like a motorcycle, the Uno tilts when turning. The control systems must maintain proper balance throughout each turn. We initially tried to develop and test the balance and tilt control loops independently, but early hardware tests showed us that they were coupled. To better understand these systems and their interdependencies, we ran simulations in Simulink. We could then examine moments of inertia, velocity, control signals, and other system characteristics that would be difficult or impossible to measure in real-world tests.

We built the basic framework by importing the mechanical design from SolidWorks® 3D CAD software into SimMechanics (Figure 2). To this framework we added the electric motor, a potentiometer, updated ball joints to link the push rods, and the mass of the rider. Via simulation, we applied motor torque and measured the tilt of the system for a range of motor angles.

![Figure 2. SimMechanics model of the tilt system.](image)

After postprocessing the simulation results in MATLAB®, we developed a transfer function between motor rotation and bike tilt. This analysis enabled us to better understand how the Uno would respond to changes in tilt motor torque (Figure 3). We used the same approach to understand the relationship between differential torque applied to the wheels and the radius of the resulting turn.

![Figure 3. Graph of the Uno tilt system step response to a tilt motor command. Green = the ideal response; blue = the simulated response.](image)
The complete tilt model (Figure 4) enabled us to experiment with different sensors. We supplemented the Inertial Measurement Unit with a potentiometer in the tilt system. Based on simulation results acquired from the Simulink model, we later moved to a higher-resolution analog-to-digital converter, which we ultimately used in the final design.

![Figure 4. Simulink model of the Uno tilt system.](image)

**Tuning Controller Gains**

BPG software engineers developed the proportional integral derivative (PID) controllers for the Uno III in C, intending to use data gathered during hardware tests to tune the controller gains. In practice, this proved to be impractical using a traditional Ziegler-Nichols method because when we increased the proportional gain, the system’s output never stabilized as we expected.

To resolve this problem, we built a simple PID controller in Simulink and ran simulations with the plant model of the tilt system. We placed scopes throughout the model to collect data, which we postprocessed in MATLAB. This analysis enabled us to first better understand the tilt system and then tune the controller gains until the system worked in simulation. We then adjusted the controller gains in our C code to match the ratio that we had verified in Simulink. The adjustment led to a breakthrough—we were able to actually ride the Uno III for the first time.

**Modifying the Hardware and Controller Software**

In addition to helping us tune controller gains, Simulink simulations also provided design insights that led to control algorithm changes and hardware modifications. For example, we ran multiple simulations in Simulink to see if gain scheduling could improve the tilt system. We found that a stepwise loop that used one set of gains for tilt angles between -3 and +3 degrees and different gain values for progressively larger tilt angle ranges produced better overall performance than a linear PID loop.

Later, we used Simulink and SimMechanics to explore mechanical changes to the system. In one instance, we ran simulations to ensure that the tilt motor had enough torque to move the 350-pound Uno and its rider from side to side. After conducting these simulations, we weren’t convinced that the motor at its current size would be capable of moving the Uno quickly enough. While the results were not definitive, we decided to err on the side of caution and use a larger motor.

Simulink simulations also helped us identify a deficiency with our analog-to-digital converter (ADC). Using some basic ADC blocks in Simulink, we built and simulated a simple model that helped us identify dead spots in our control algorithm that were affecting performance. To address the problem, we replaced the hardware ADC on the Uno with one that had a resolution four times higher.

**Simulating the Transition and Power Systems**

Because the Uno’s transformation from dicycle to motorcycle occurs while the vehicle is moving, ensuring the safety of this transition is a top priority. To simulate the transition with Simulink and SimMechanics, we combined a simple model of the balance control system with an inverted pendulum model. We then estimated the position of the Uno’s center of mass relative to the traction wheels at several states in the transition process, and used Simulink to verify that the mechanical model was controllable for each state.
As the Uno moves from prototype to preproduction, we are expanding our use of Simulink to model and simulate aspects of the Uno that would be too costly, dangerous, or time-consuming to experiment with on the actual hardware. We recently used SimPowerSystems™ to model the Uno’s 48V power system, including the batteries, switches, and motors, to capture inductance and capacitance effects. Instead of hooking up probes and meters to the actual power system, we simulated it to track down the source of the spikes that we had identified during testing.

Going forward, we plan to reuse the model for additional reliability checks and to estimate battery life for various drive cycles and conditions.

### Products Used
- MATLAB
- Simulink
- SimMechanics
- SimPowerSystems
- Simscape

### Learn More
- Video: Origins of the Uno (8:01)
- Video: Importing and Merging CAD Models into SimMechanics (7:13)

See more articles and subscribe at mathworks.com/newsletters.