

Segway LLC Delivers Innovative Transporter Using MathWorks Tools

The Segway™ Human Transporter (HT) is the first two-wheeled, electric-powered personal transportation device capable of emulating human balance. A breakthrough technology called Dynamic Stabilization balances the Segway HT—whether the user is traveling at 10 mph, carrying a heavy load, maneuvering in tight spaces, or standing still.

The Segway HT concept was developed at DEKA, a product development company owned and run by inventor Dean Kamen. After years of lab testing at DEKA, Kamen founded Segway LLC to develop and commercialize the new transporter.

Segway engineers used MATLAB®, Simulink®, and toolboxes and blocksets to refine their conceptual designs and deliver a reliable, durable commercial product.

“MATLAB enabled us to get rapid, visual representations of data, which were invaluable for solving critical design problems,” says John Morrell, Dynamics and Software Manager at Segway. “The MathWorks tools helped our engineering teams work within an iterative process, quickly simulating, designing, and verifying system characteristics.”

THE CHALLENGE

Because this product was the first of its kind, Segway needed versatile tools that could solve engineering problems not encountered in automobile or other vehicle designs.

The production engineering team had to ensure that the Segway HT’s dynamic and mechanical performance would meet strict criteria for durability, reliability, performance, and battery and capacitor life. In particular, they



The Segway Human Transporter.

needed to verify the fault detection and response system, study the effects of drag torque, and verify the inertial balancing system.

Implementing the microprocessor that would balance and control the Segway HT was an additional challenge, as the microprocessor chosen had only fixed-point mathematics capability and limited memory.

THE SOLUTION

To save time, multiple engineering groups at Segway worked on various parts of the Segway HT simultaneously. MathWorks tools provided a common platform for sharing models and testing results across the teams.

“MathWorks tools shortened development time and increased our confidence in the design,” says Morrell. “Everyone here uses MATLAB and loves it.”

The Dynamics Development team used MATLAB, Simulink, the Control System Toolbox, and the Fixed-Point Blockset to model the inertial sensing system and design state-estimation algorithms for the primary control loops.

THE CHALLENGE

To design, deliver, and manufacture the first electric-powered, dynamically stabilized transportation machine

THE SOLUTION

Use MathWorks tools to refine early conceptual designs and deliver a reliable, durable commercial product

THE RESULTS

- Development cycles accelerated
- Microprocessor memory constraints met
- Lifecycle design goals confirmed

“MATLAB is an integral part of our product design process. It allows us to make the technical story clear so that we all understand how the system is operating.”

John Morrell, Segway

They simulated the system in floating point and fixed point and then ran system data into the simulations and compared the real system results to the simulated results. Using Simulink, the Signal Processing Toolbox, and the DSP and Fixed-Point blocksets, the engineers modeled filters and tested them to ensure that they could handle sensor failures. They then used the models to run the algorithms to handle noise and failure modes.

The Motor Drive Development team used MATLAB to model the motor and electronic drive components. The thermal characteristics of the system and high-speed motor drive dynamics were modeled, verified, and reduced to simpler models for implementation in the system. Using MATLAB scripts, engineers distilled data derived from real system tests. The test results helped them modify the motor drive control algorithm, reducing current ripple and increasing capacitor life.

“The MATLAB script was very fast, did not use huge amounts of disk space, and provided excellent quality and visualization in the graphs,” says J.D. Heinzmann, who led the Motor Drive Development team. “No other tools could enable us to accomplish this.”

To ensure that the mathematics would work correctly when implemented on the Segway HT microprocessor, the motor drive engineers used MATLAB and Simulink to analyze polynomial approximations to fixed-point trigonometric functions.

Simulink helped the motor drive engineers model the effects of drag torque caused by simulated shorting and motor failure.

Dynamics engineer David Robinson visualized and tested motor and battery design configurations with MATLAB. Once the motors and batteries were selected, Robinson used

MATLAB to design estimation algorithms for real-time execution in the production processor. After developing the algorithms in floating point, he tested them in fixed point and confirmed that they would run correctly on the final Segway HT processor.

“The advanced visualization capabilities of MATLAB let us view all the data at once,” says Robinson. “This helped us reach conclusions quickly and determine the best approach to model the system and transfer learning of our findings throughout Segway.”

THE RESULTS

■ **Development cycles accelerated.** Using MATLAB and Simulink, the engineers simulated, designed, and verified the system characteristics, enabling them to learn, make changes, and validate quickly.

■ **Microprocessor memory constraints met.** Using MATLAB, the engineers coded tight algorithms that handle the operation of 12 motor switches, machine monitoring, and balance during operation—all within the 32K microprocessor limit.

■ **Lifecycle design goals confirmed.** By reading live ride data into MATLAB and using a fatigue-counting algorithm, engineers confirmed that the life of the component parts would meet the five-year requirement.

To learn more about Segway, visit www.segway.com

APPLICATION AREAS

- Algorithm development
- Data analysis
- Modeling and simulation
- Signal processing
- Testing

PRODUCTS USED

- MATLAB
- Simulink
- Control System Toolbox
- DSP Blockset
- Fixed-Point Blockset
- Signal Processing Toolbox

www.mathworks.com