

Using MATLAB, Virtual Reality, and a Treadmill to Investigate How Humans Use Visual Input to Control Their Gait

By Shawn O'Connor, Simon Fraser University

Send email to [Giampiero Campa](mailto:giampiero.campa@sfu.ca)

Unless they are in a hurry, people rarely think about how fast they're walking. Instead, they select a gait that feels natural. Researchers have found that this gait is *energetically optimal*—it enables the walker to expend the least amount of energy when covering a given distance. While scientists have known for some time that people walk at a pace that minimizes energetic cost, they have yet to determine how key sensory inputs, such as metabolic rate and vision, affect natural gait selection.

At Simon Fraser University, my colleagues and I used MATLAB® and Simulink® to measure the effect of vision on walking speed adjustments. In addition to increasing scientists' understanding of why people walk as they do, this type of research provides specific knowledge of how gaits are chosen, with implications for devising rehabilitation therapies that encourage effort and patient participation.

Experimental Setup

Our experiment hinged on the ability to alter test subjects' visual perception of their walking speed. The setup consisted of a specially instrumented, split-belt treadmill; a curved projection screen; a Vicon motion-capture system; and a Simulink based controller (Figure 1).

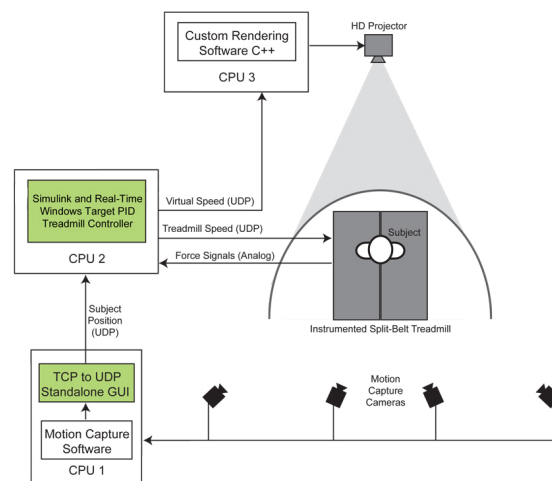


Figure 1. The experimental setup.

We positioned the treadmill directly in front of the screen (Figure 2), onto which we projected a view of a moving virtual hallway (Figure 3). The motion-capture system detected the subject's position on the treadmill. The controller used this information to manage the treadmill's speed: When the subject moved forward, the controller increased the treadmill speed to bring him back to center. When the subject moved backward, the controller recentered him by slowing the treadmill speed. The controller also managed the images displayed on the projection screen based on the speed of the treadmill and the subject's position. Most importantly, the controller enabled the research team to change the speed of the projected virtual hallway relative to the walking speed of the subject. We could then quantify the subjects' dynamic response to the perceived change.

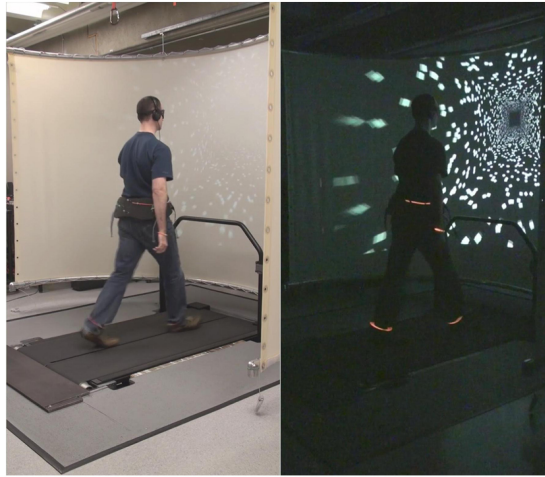


Figure 2. The experimental setup with lights turned on to show the treadmill and projection screen (left) and with lights turned off to show the virtual hallway (right).

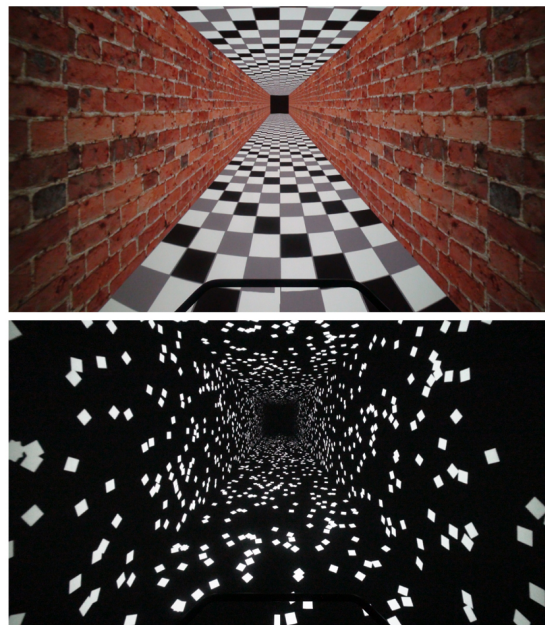


Figure 3. Two versions of the virtual hallway.

We had two implementation options for the controller: We could build it ourselves or purchase a third-party custom solution. The custom solution that we evaluated had an estimated cost in the hundreds of thousands of dollars, and it provided no way for us to make the software changes that are inevitable in experimental research.

Developing the control system in MATLAB and integrating it with a \$5,000 virtual reality system enabled us to stay within budget and gave us full control over the software. This control proved vital, as our initial experiments revealed minor design challenges that we had not considered when creating the system.

Developing the Controller

The Simulink control-system model incorporates a proportional-integral-derivative (PID) controller (Figure 4) and runs in real time on a standard PC using Real-Time Windows Target™. It receives data on the subject's position from the motion-capture system and produces commands to control the speed of the treadmill and of the hallway images presented by the virtual reality system.

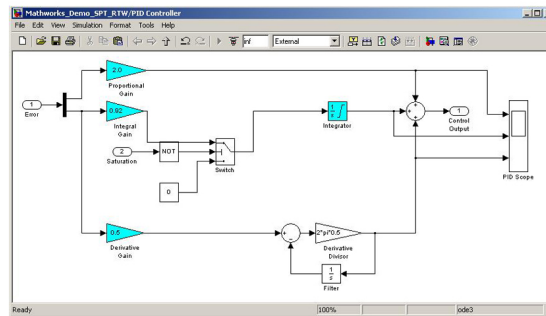


Figure 4. Simulink PID controller for calculating treadmill and hallway-image speed.

In addition to the PID controller, the complete Simulink controller model includes several safety features, including a speed limiter and an emergency stop (Figure 5). A key component is the Speed Multiplier block, which enables us to alter the relationship between the virtual hallway speed and the treadmill speed. By default, the controller manages the hallway speed so that the hallway appears to be moving past the subject at a speed corresponding to his walking speed. Using the Time Varying Speed Ratio block, we can add a step function to suddenly increase or decrease the speed of the hallway relative to the treadmill. We can also add more complex sine wave functions that gradually increase and decrease the relative hallway speed.

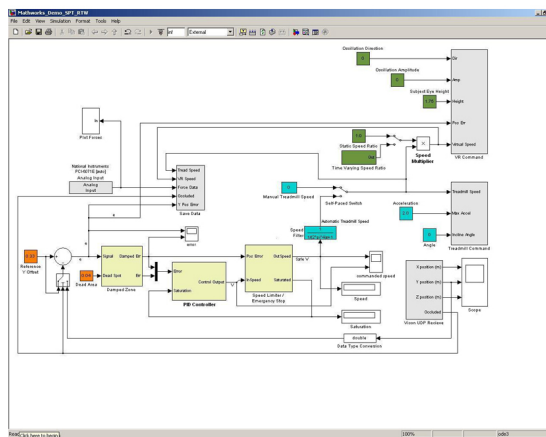


Figure 5. The complete Simulink controller model.

Integrating the Motion-Control System and the Controller Model

Before we could test the Simulink controller model on our setup, we had to link it to our motion-control system, which streams position data using Transmission Control Protocol (TCP). Real-Time Windows Target supports standard network connections using User Datagram Protocol (UDP), which meant that we needed to convert TCP segment data to UDP packets.

Using MATLAB and Communications System Toolbox™, we wrote a simple program that reads TCP data and relays it over the network using UDP. We then created a standalone version of the program using MATLAB Compiler™. The program includes an interface that can be used to configure the necessary IP addresses and ports. We ran this standalone program directly on the motion-capture PC, which did not have MATLAB installed.

Adjusting the Design to Improve Validity

While running experiments with actual test subjects, we discovered several deficiencies in our initial setup. Instead of waiting weeks for another company to implement fixes, we could rapidly implement them ourselves—often in a single day—because we had built the controller ourselves in Simulink rather than buying a custom-built solution. Because the experiment relies so heavily on human perception, the ability to test the setup ourselves, rapidly make minor adjustments to the controller, and then retest was invaluable.

The validity of our results depended on our ability to make our subjects feel as if they were walking down a real hallway. We soon discovered two problems that affected our ability to immerse the subjects in the virtual hallway experience and required changes to the controller design. First, our initial design was actually too sensitive, because the treadmill started moving as soon as the subject moved away from its center point. As a result, the subjects frequently felt the treadmill speed changing. To mitigate this effect, we changed the Simulink model to implement a *dead zone*—an area in the center of the treadmill in which the subject could move without making the treadmill speed up or slow down.

Soon afterward, we determined that as the subject neared the projection screen, the hallway appeared to move faster, even though we had not changed its speed. This necessitated another change to the model to adjust the speed of the hallway based on the subject's proximity to the screen. Taken together, these minor changes made the subject's experience more realistic and more fluid, increasing the validity of our results.

Modeling the Dynamics of Human Perception

Once we had refined the experimental setup, we conducted numerous trials with multiple subjects, using MATLAB and Statistics Toolbox™ to analyze results and conduct hypothesis testing. We measured how long, on average, it took subjects to respond to a step function perturbation in the perceived hallway speed. For sine wave perturbations, we measured the detected gain on the sine wave output.

Essentially, we took a systems dynamics approach in which we supplied known inputs to the system—a step function or a sine wave—and measured the response output. We then processed this input-output data with System Identification Toolbox™ to construct a mathematical model of a human being as a dynamic system that uses visual information to adjust walking speed. The model enabled us to quantify the dynamics of the processes at work and better understand the time constants involved.

We are currently using a similar experimental setup to measure the effect of blood gas levels on gait selection. In this experiment we use a Simulink model and Real-Time Windows Target to control the amount of oxygen and carbon dioxide available to our treadmill walking subjects, which in turn affects the level of these gases in their blood. We will test whether blood gas levels are a key metabolic indicator by controlling oxygen and carbon dioxide levels as a function of measured gait parameters, such as speed and step frequency, and testing if subjects shift their preferred gait to maximize blood oxygen and minimize carbon dioxide. This experiment will provide a better understanding of the sensory inputs that people use to choose gaits, helping therapists maximize patient effort during rehabilitation.

Products Used

- [MATLAB®](#)
- [Simulink®](#)
- [Communications System Toolbox™](#)
- [MATLAB Compiler™](#)
- [Real-Time Windows Target™](#)
- [Statistics Toolbox™](#)
- [System Identification Toolbox™](#)

Learn More

- [PID Control Design Made Easy](#)
- Video: [Visualizing Data](#)

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