In researching prior work in driving simulation, we have come across several articles that may have a severe influence on our approach to building Dr. Ferris’ predictive correction simulator for Volkswagen. Since this is an ongoing project, with publication being an eventual goal, Dr. Ferris and several of his graduate students have provided us their initial draft paper on their work with predictive correction which we have used to attain a grasp on where they currently are in the project. Currently their technical work resides exclusively in the MATLAB and Simulink systems, with some integration with CarSim (Matthews, Cho, Ferris, Schlinkheider). However, due to project goals and the abilities of CarSim being in conflict (CarSim cannot handle interactive input), alternative methods are required for their project to continue.

Due to 3D simulation being increasingly in-demand, several research teams have begun integrating high-quality 3D game engines to display their simulation data. These game engines, namely Unity 3D and the Unreal Development Kit, come with free non-commercial licenses, which are ideal for academic research. Additionally, these game engines typically host physics engines as well, which is often used by these simulations (Cai, Sun 439-447) (Feng, Shang, Yang). In cases where business logic for simulators cannot be done in the simulator itself, game engines can still be used to represent only the virtual world itself. Simulations run in MATLAB can communicate with these engines through non-standard means, usually TCP sockets or shared files, to direct the camera in the game engine where to look (Hatnik, Altmann) (Valldeperas).

One of the most relevant new technologies we have come across is a software suite known as rFactor. rFactor naturally integrates with MATLAB and Simulink, and provides a 3D graphics engine that appears to be on par with that of Unity3D or Unreal (Koskela, Nurkkala, Kalermo, Järvilehto). rFactor is also capable of providing incredibly detailed vehicle analytics, as it was designed to assist engineers involved in automobile racing. Additionally, rFactor also has native support for Logitech racing input controls, which provides a realistic input to the program, as well as haptic feedback (resistance, jerks, or rumble) to the user (Dempsy, Fish, Picarelli).

**BIBLIOGRAPHY**

Browne, M, T Anderson, A Enmark, D Moraru, and A Shearer. "Parallelization of MATLAB for

Euro50 integrated modelling." *Modeling and Systems Engineering for Astronomy*.

(2004): n. page. Print.

<<http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=848287>>.

This source illustrated how to use multi-threading in MATLAB to decrease execution time and provide a more real-time data set. This method can be used as an optimization to accelerate calculations we will be needing for our simulator.

Cai, Xingquan, and Limei Sun. "Low-Cost Virtual Driving Simulator Design and Its

Application." *Lecture Notes in Electrical Engineering*. 212. (2013): 439 - 447. Print.

<<http://link.springer.com/chapter/10.1007/978-3-642-34531-9_46>>.

This article follows a team, similar to ours, constructing a driving simulator with similar constraints to ours. This team chose to use a Logitech controller as a realistic user input tool, along with providing haptic feedback for realism. Additionally, the simulation engine itself is built entirely in Unity, demonstrating the possibility of an all-Unity build.

Dempsy, Mike. Fish, Garron. Picarelli, Alessandro. “Using Modelica models for

Driver-in-the-loop simulators.” The 9th International Modelica Conference, September

3-5, 2012, Munich, Germany. Leamington Spa, UK: Edmund House, 2012.

This article is relevant because it gives a good general overview of the entire simulation process. This article suggests using “Modelica” as a possible development choice. This article also gives another example of using “rFactor” for a research driving sim.

Feng, Xu, Shang Daguo, and Yang Hongchen. "Simulation Research of Crime Scene Based

on UDK." (2010): n. page. Print.

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5690391>>.

This article shows a further use of game engines in computerized simulation. While Unity may end up being our 3D engine of choice, this team showed how to build a fully implemented simulator using only the Unreal Development Kit and its internal tools in a very similar fashion we would use if we chose to build an exclusively Unity-driven simulator.

Hatnik, Uwe, and Svenn Altmann. "Using ModelSim, Matlab/Simulink and NS for Simulation

of Distributed Systems." *Parallel Computing in Electrical Engineering*. (2004): 114 -

119. Print. <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1376743>>.

This source demonstrated how to link MATLAB/Simulink with other software packages for simulations involving multiple software packages simultaneously. Best methods involved either sharing common resource/data files or interacting via local TCP sockets. This will be relevant for us if we end up using a virtual environment that does not natively support MATLAB.

Kalermo, J. Nurkkala V-M. Koskela, K. Järvilehto, T. “Driving Characteristics and

Development of Anticipation of Experienced and Inexperienced Drivers When

Learning a Route in a Driving Simulator.” Driving Simulation Conference September

6-7, 2012, Paris, France. Kajaani, Finland: Kajaani University of Applied Sciences,

2012.

This article discuses the use of the same “rFactor” based sim from Finland that was used to study driver behavior. The article is relevant because it also suggests design ideas for courses that may help in designing a course to highlight our added functionality.

Koskela, K., V-M. Nurkkala, J. Kalermo, and T. Järvilehto. "Low-cost Driving Simulator for

Driver Behavior Research." (n.d.): n. pag. *Google Scholar*. Web. 20 Sept. 1994.

<<http://weblidi.info.unlp.edu.ar/worldcomp2011-mirror/CGV4545.pdf>>.

This article is relevant because it gives more details for creating relatively cheap and accurate driving sims. This source also gives more support for using “rFactor” and “Bob’s Track Builder” to create the sim.

Li, S., L. He, “Co-simulation Study of Vehicle ESP System Based on ADAMS and MATLAB” *Journal of Software*, Volume 6, No. 5, 2011 pp 866-872.

This article describes a situation very much like our own project, where a vehicle simulation model is being created to test a possible upcoming technology. In this case, it is experimenting with Electronic Stability Programs which selectively apply brakes to individual wheels of the car in order to correct its course.

Manoj, K., B. Steward, A. Kelkar, Z. Kemp II “Modeling and real-time simulation architectures for virtual prototyping of off-road vehicles“ *Virtual Reality* Volume 15, Issue 1, pp 83-96. Spriger-Verlag 2011

This article is highly related to our project goals, as the authors are modeling a realistic environment for a real-time simulation of a vehicle. This article also goes into great depth about how the model, visualization, and hardware are linked, which is a major part of our duties in this project team.

Matthews, C., Cho, S., Ferris, J., Schlinkheider, J. et al., "Using Performance Margin and

Dynamic Simulation for Location Aware Adaptation of Vehicle Dynamics," *SAE Int. J.*

*Passeng. Cars - Mech. Syst.* 6(1):2013

This article was published by the team we are working with on our project. It gives us an idea of the work that has already been done on this project, as well as giving us a good conceptual understanding of what we are working on

McDonald, David. “Electric Vehicle Drive Simulation with MATLAB/Simulink.” North-Central

Section Conference March 23-24 2012, Ohio Northern University, OH. Sault Ste

Marie, MI: LSSU, 2012.

This source was relevant in the way that it sheds light on aspects needed to model a vehicle engine in MATLAB/Simulink. It is an electric engine and not part of a fully interactive simulation, but the Model/MATLAB insights could still prove valuable.

Shala, A., and R. Likaj. "Fuzzy Logic Control And 3D Simulation Of Road Vehicle." *Annals Of DAAAM & Proceedings* (2010): 1457-1458. *Academic Search Complete*. Web. 24 Sept. 2013.

This article describes another possible visualization method for vehicle simulation, Virtual Reality Markup Language (VRML).

Valldeperas, Roger. "Production Cell Simulation Visualization in 3D." (2013): n. page. Web. 24

Sep. 2013. <<http://www.diva-portal.org/smash/get/diva2:639620/FULLTEXT01.pdf>>.

This article presented a similar simulator design challenge that is similar to ours, and chose to adapt the use of game engines as the medium of visualization. The article compared the two leading game engines, Unity and Unreal, in a context similar to how we will be using them, and chose Unity. Similarly, it demonstrated the use of TCP sockets in Unity for receiving virtual input.

Weinberg, Garrett. Harsham, Bret. “Developing a Low-Cost Driving Simulator for the

Evaluation of In-Vehicle Technologies.” the First International Conference on

Automotive User Interfaces and Interactive Vehicular Applications, Sep 21-22, Essen, Germany. Cambridge, MA: Mitsubishi Electric Research Labs, 2009.

This article is relevant because it is a strong proof of concept of developing a low cost driving simulator to test a potential hardware edition for automotives. This resource also suggested the use of “rFactor” and “Bob’s Track Builder” (which also supports the Logitech Force-Feedback controllers) as a possible front-end to the Simulation.

A.G. Wood, L.J. Mountain, R.D. Connors, M.J. Maher, K. Ropkins “Updating outdated predictive accident models” *Accident Analysis & Prevention*, Volume 55, June 2013, Pages 54–66

This article describes different predictive accident models and how they can become out of date, and recommends a way to keep them up to date. While our group is not directly responsible for the predictive model, this article would be good to share with the rest of the team and to keep in mind for the project.