

Simulation System to Aid in Vehicle Simulator Design

T Kader^a, R Stopforth^b, G Bright^c

Received 4 February 2016, in revised form 23 August 2016 and accepted 18 October 2016

Vehicle simulators aim to provide the user with a driving experience that replicates the sensations experienced in a real vehicle. An important aspect is the replication of motion cues. The replication of motion cues in the vehicle simulator is limited by platform workspace and the selected position control systems performance. In this work a Matlab/Simulink setup that incorporates the various aspects involved in replication of motion sensations was created. This includes input data scaling, classical washout algorithm and inverse kinematic implementations. By adjusting the input data scaling and classical washout algorithm filter parameters platform motion that adheres to the actuator motion constraints was assured. Therefore this simulation system makes the design process easier since no parameter adjusting was done on the actual motion platform. This simulation system also ensures safe operation of the vehicle simulator system for the user and ensures safety of the vehicle simulator mechanical structure. Additionally the system allowed for the derived inverse kinematic equations to be verified and evaluated the effectiveness of the classical washout motion cueing algorithm.

Additional keywords: Parallel manipulator, Inverse kinematics, Motion cueing, Vehicle simulator

1 Introduction

A vehicle simulator makes use of a parallel manipulator to provide motion cues to the user. Parallel manipulators consist of a fixed base and a number of kinematic chains connected to a moving platform [1]. These manipulators have a large load carrying capacity which makes them ideal for usage in a vehicle simulator. Gough developed the first parallel manipulator used as universal tyre test machine [2]. Stewart developed a six-degree of freedom parallel manipulator to be used as a flight simulator [3]. Parallel manipulators have fairly straightforward inverse kinematics but have complex forward kinematics. Due to complex forward kinematics and difficulty to manufacture spherical joints at low cost [4], the development of lower degree-of-freedom parallel manipulators has been researched extensively. Lee and Shah [5] presented closed form solutions for the forward and inverse kinematics of a 3-DOF parallel manipulator.

- Discipline of Mechanical Engineering, Mechatronics and Robotics Research Group (MR2G), University of KwaZulu Natal, South Africa, taahirkader@gmail.com
- Discipline of Mechanical Engineering, Mechatronics and Robotics Research Group (MR2G), University of KwaZulu Natal, South Africa, stopforth.research@gmail.com
- Discipline of Mechanical Engineering, Mechatronics and Robotics Research Group (MR2G), University of KwaZulu Natal, South Africa, brightg@ukzn.ac.za

Vehicle simulators first originated from the development of flight simulators. In the early 1980's Daimler-Benz created a high fidelity vehicle simulator [6]. Subsequent simulators have been created by General Motors [7], University of IOWA [8], and Toyota [9]. Vehicle driving was a task thought to be mainly dominated by visual information. In recent times it was shown that other sensory information, such as the vestibular system contribute to motion perception [10]. The vestibular system consists of the semi-circular canal and otolith. The otolith senses linear acceleration, via specific force, and head tilt relative to gravity. The semi-circular canal senses angular velocity. Vehicle simulators aim to replicate the motion sensations experienced in an actual vehicle. The problem is that the motion platform workspace is limited; this makes it difficult to recreate the motion sensations. Various motion cueing algorithms have been developed to try and replicate the sensations felt in a real vehicle [11]; these include the classical washout algorithm [12], adaptive washout algorithm [13], optimal washout algorithm [14] and model predictive control [15].

Commercial usage of vehicle simulators has increased due to lower costs and improved fidelity. A vehicle simulator was used in commercial truck driver training, to teach improved driving techniques which increased fuel efficiency [16]. The Arizona Department of Transportation (ADOT) used a vehicle simulator to train snowplough operators to react to potential hazards and improve fuel efficiency with improved driving techniques [17]. Mining simulators are now also used extensively to train operators in a safe and realistic environment and many mining companies now see the value in purchasing such systems.

SimMechanics is a toolbox that provides a multibody simulation environment which allows the modelling and simulation of mechanical systems using their geometric layout and structural properties [18]. Yu et al. [19] developed a simulation system in Matlab/Simulink, using the SimMechanics toolbox, to verify the inverse kinematic equations for a 3-DOF motion platform. Motion simulators operate with position control systems that have limited performance. In the current vehicle simulator, illustrated in figure 1, the motion platform is a 3-DOF platform that is driven by three linear pneumatic actuators. These actuators have motion constraints that are imposed on them. It is important to adhere to these constraints to prevent mechanical damage of the vehicle simulator and to ensure safety of the driver. Many formatting features of this journal have developed over decades of mostly the printed version and present policy is to honour this tradition by minimizing deviations from this formatting. The word processing program of choice remains Microsoft Windows. The editorial/copy editing process is integrated, so that as the final MS-Word version emerges, it merely needs to be converted to .pdf file format for publishing on the internet.



Figure 1: Vehicle Simulator Model

The research contributions of this paper include:

- Developing a simulation system, in Matlab/Simulink, that incorporates the vehicle input data scaling and classical washout algorithm implementation into the inverse kinematic analysis simulation system developed in previous work [19,20].
- Using the developed simulation system to take in the vehicle data directly and analyse the output from the structural model of the motion platform to determine if the motion platform adheres to the motion constraints imposed. By incorporating the vehicle input data scaling and classical washout algorithm into the simulation system, the parameters of these systems are adjusted easily in the simulation environment until the actuator motion constraints are adhered to. Once reasonable

and safe performance is achieved the system can be implemented on the actual 3-DOF motion platform.

- Additionally the developed simulation system is used to evaluate the classical washout algorithm and to verify the derived inverse kinematic equations. The developed simulation system is applicable to all commercial vehicle simulators that make use of parallel manipulators.

2 Kinematic Analysis

The following section provides the inverse kinematic analysis for the 3-DOF motion platform used in the vehicle simulator. Motion for the platform is specified with three independent end-effector parameters; these include the translational motion along the z-axis (heave) and the rotational motion about the x-axis (roll) and y-axis (pitch) respectively. Figure 2 illustrates the motion platform for the vehicle simulator with the various coordinate systems. Coordinate frame $\mathbf{A}(x, y, z)$ is attached to the centroid, \mathbf{O} , of the base of the motion platform and coordinate frame $\mathbf{B}(u, v, w)$ is attached to the centroid, \mathbf{P} , of the moving platform.

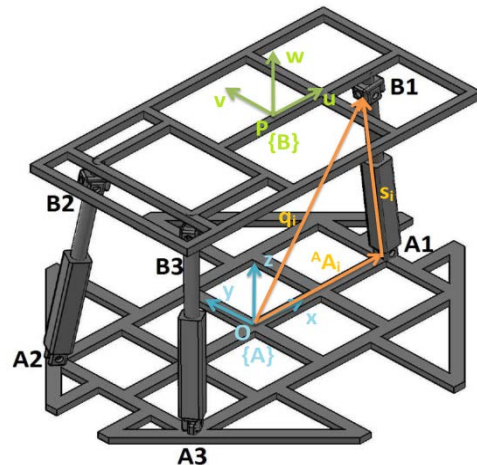


Figure 2: 3-DOF Motion Platform with Coordinate Systems

The x-y plane contains revolute joints $A_i, i = 1$ to 3 , and the u-v plane contains universal joints $B_i, i = 1$ to 3 . The linear actuators used in the system are themselves kinematic constraints, these actuators consist of cylindrical joints designed to provide linear motion in the actuator axis direction and rotational motion about this axis. The motion of a point on the moving platform can be fully expressed in the fixed base coordinate system \mathbf{A} , by a translational component and a rotational component. To describe this motion some parts of the motion has to be transformed from one coordinate system to another coordinate system. Unit vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$, are defined along the u, v, w axes of the moving coordinate system \mathbf{B} . The rotation matrix from coordinate frame \mathbf{B} to coordinate frame \mathbf{A} is defined as:

$${}^A R_B = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \quad (1)$$

The vector ${}^A A_i = [a_{ix} \ a_{iy} \ a_{iz}]^T$ is the position of the revolute joint A_i with respect to the frame \mathbf{A} and vector ${}^B B_i = [b_{iu} \ b_{iv} \ b_{iw}]^T$ is the position of the universal joint B_i with respect to the frame \mathbf{B} .

The three revolute joints, in the base coordinate frame \mathbf{A} , are located at the following coordinates in metres ${}^A A_1 = [0.6 \ 0 \ 0]^T$, ${}^A A_2 = [-0.6 \ 0.25 \ 0]^T$ and ${}^A A_3 = [-0.6 \ -0.25 \ 0]^T$.

The three universal joints, in the moving platform coordinate frame \mathbf{B} , are located at the following coordinates in metres ${}^B B_1 = [0.5 \ 0 \ 0]^T$, ${}^B B_2 = [-0.5 \ 0.15 \ 0]^T$ and ${}^B B_3 = [-0.5 \ -0.15 \ 0]^T$.

Point \mathbf{P} , the centroid of the moving platform, with respect to fixed base frame \mathbf{A} is represented by ${}^A P = [p_x \ p_y \ p_z]^T$.

The position vector q_i of B_i with respect to coordinate frame \mathbf{A} is given by the following transformation:

$$q_i = {}^A P + {}^A R_B {}^B B_i \quad (2)$$

The motion of each limb is constrained by the revolute joints, which attaches the limb to the fixed base. The motion is constrained in one of the following three planes:

$$q_{1y} = 0 \text{ for } i = 1$$

$$q_{2y} = -\frac{0.25}{0.6} q_{2x} \text{ for } i = 2$$

$$q_{3y} = \frac{0.25}{0.6} q_{3x} \text{ for } i = 3$$

Expanding the above motion constraint results by using the position vector q_i of B_i with respect to coordinate frame \mathbf{A} for $i = 1$ to 3 gives the following:

$$p_y + 0.5u_y = 0 \quad (3)$$

$$p_y - 0.5u_y + 0.15v_y = -\frac{0.25}{0.6}(p_x - 0.5u_x + 0.15v_x) \quad (4)$$

$$p_y - 0.5u_y - 0.15v_y = \frac{0.25}{0.6}(p_x - 0.5u_x - 0.15v_x) \quad (5)$$

On the platform a rotation about the x-axis (roll) is represented by α , a rotation about the y-axis (pitch) is represented by β and a rotation about the z-axis (yaw) is represented by γ . The following short form representations are used throughout the rest of this paper, $\sin \blacksquare = s\blacksquare$, $\cos \blacksquare = c\blacksquare$, $\tan \blacksquare = t\blacksquare$ and $\sec \blacksquare = s\blacksquare$. The rotation matrix for the platform is subsequently given by:

$${}^A R_B = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

$${}^A R_B = \begin{bmatrix} c\gamma c\beta & c\gamma s\beta s\alpha - s\gamma c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha \\ s\gamma c\beta & s\gamma s\beta s\alpha + c\gamma c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha \\ -s\beta & c\beta s\alpha & c\beta c\alpha \end{bmatrix} \quad (6)$$

Adding Eq. (4) and Eq. (5) and substituting with values from Eq. (6) the motion constraint for the y-axis translational motion is given by:

$$p_y = 0.5s\gamma c\beta - 0.0625c\gamma s\beta s\alpha + 0.0625s\gamma c\alpha \quad (7)$$

Subtracting Eq. (5) from Eq. (4) and substituting with values from Eq. (6) the motion constraint for the x-axis translational motion is given by:

$$p_x = 0.5c\gamma c\beta - 0.36c\gamma c\alpha - 0.36s\gamma s\beta s\alpha \quad (8)$$

Substituting $u_y = s\gamma c\beta$, which follows from Eq. (1) and Eq. (6), into Eq. (3) results in the following motion constraint for rotational motion about the z-axis:

$$\gamma = \sin^{-1}\left(-\frac{2p_y}{\cos\beta}\right) \quad (9)$$

From figure 2 the leg vector $s_i = [s_{ix} \ s_{iy} \ s_{iz}]^T$ with respect to frame \mathbf{A} , is given by:

$$s_i = {}^A P + {}^A R_B {}^B B_i - {}^A A_i \quad (10)$$

The magnitude of each leg vector gives the leg length of each leg. Taking the magnitude of each leg vector s_i gives the following leg length equations for $i = 1$ to 3:

$$l_1 = \sqrt{(p_x + 0.5c\gamma c\beta - 0.6)^2 + (p_y + 0.5s\gamma c\beta)^2 + (p_z - 0.5s\beta)^2} \quad (11)$$

$$l_2 = \sqrt{(p_x - 0.5c\gamma c\beta + 0.15c\gamma s\beta s\alpha - 0.15s\gamma c\alpha + 0.6)^2 + (p_y - 0.5s\gamma c\beta + 0.15s\gamma s\beta s\alpha + 0.15c\gamma c\alpha - 0.25)^2 + (p_z + 0.5s\beta + 0.15c\beta s\alpha)^2} \quad (12)$$

$$l_3 = \sqrt{(p_x - 0.5c\gamma c\beta - 0.15c\gamma s\beta s\alpha + 0.15s\gamma c\alpha + 0.6)^2 + (p_y - 0.5s\gamma c\beta - 0.15s\gamma s\beta s\alpha - 0.15c\gamma c\alpha + 0.25)^2 + (p_z + 0.5s\beta - 0.15c\beta s\alpha)^2} \quad (13)$$

The leg length equations determine the actuator stroke lengths for a particular trajectory of the end-effector.

3 Motion Cueing

The classical washout algorithm is designed to replicate the motion sensations felt in a vehicle without breaching the platform workspace constraints. Implementation is divided into two channels, the first channel is used for translational motion and the second is used for rotational motion. Inputs to the motion cueing strategy are the specific force vector \vec{f}_{veh} and angular velocity vector $\vec{\omega}_{veh}$ experienced in the vehicle. The algorithm aims to replicate these signals, within the motion simulator at the centroid of the moving platform coordinate system \mathbf{B} , as closely as possible.

Figure 3 illustrates the translation channel for the classical washout algorithm. The translational channel is used to replicate the transient component of the vehicle specific force vector \vec{f}_{veh} .



Figure 3: Translational Channel for the Classical Washout Algorithm

\vec{f}_{veh} is first scaled to help constrain platform motion. The scaled vector \vec{f}_1 is used to generate the acceleration vector \vec{a}_1 for the centroid of the moving platform \mathbf{B} . Reid and Nahon²¹ showed that the gravitational vector signal \vec{g}_1 is given by:

$$\vec{g}_1 = {}^A R_B^T \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (14)$$

The vector \vec{a}_1 is then transformed into coordinate frame \mathbf{A} , this is done by multiplying \vec{a}_1 by the rotation matrix ${}^A R_B$. Filtering is then performed on \vec{a}_2 to filter out sustained accelerations and perform washout. The washout process is used to return the motion platform back to the neutral (centre) position. This helps in preventing steady state motion errors on the actuator legs. The output acceleration vector ${}^A \vec{a}$ is then integrated twice to produce the platform

position vector ${}^A P = [p_x \ p_y \ p_z]^T$, in the inertial coordinate frame \mathbf{A} .

Figure 4 illustrates the rotational channel for the classical washout algorithm. The rotational channel is composed of two parts, which together produce the rotation (roll-pitch-yaw) angles for the motion platform.

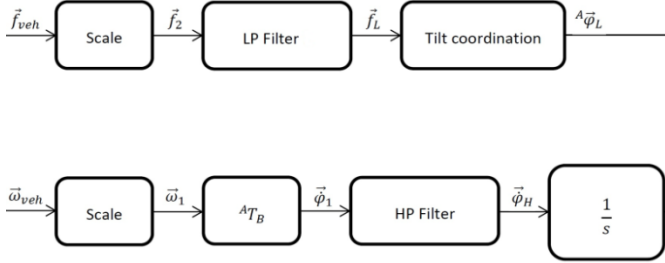


Figure 4: Rotational Channel for the Classical Washout Algorithm

The first part is called tilt coordination. Tilt coordination is used to replicate the sustained component of the vehicle specific force vector \vec{f}_{veh} via tilt of the motion platform. It aims to use a component of the gravity vector to simulate a sustained specific force. This is interpreted by the otolith as a sustained linear acceleration. The process starts with the signal \vec{f}_{veh} which is scaled and passed through a low-pass filter, this extracts the low frequency vector \vec{f}_L . In the absence of rotational motion from the vehicle angular velocity vector $\vec{\omega}_{veh}$, the roll-pitch-yaw angles can be represented by:

$${}^A \vec{\varphi} = {}^A \vec{\varphi}_L \quad (15)$$

For a simulator with small rotation angles the following approximations, from Reid and Nahon²¹, are used $f_L^z \approx g$, $\sin x \approx x$ and $\cos x \approx 1$. The roll-pitch-yaw angles based on the sustained specific force vector \vec{f}_L are subsequently approximated by:

$$\alpha_L \approx \frac{f_L^y}{g} \quad (16)$$

$$\beta_L \approx -\frac{f_L^x}{g} \quad (17)$$

$$\gamma_L = 0 \quad (18)$$

The second component of the rotational channel is used in the replication of the transient component of the vehicle angular velocity vector $\vec{\omega}_{veh}$. $\vec{\omega}_{veh}$ signal is scaled to ensure the platform rotational motion limits are not reached. The scaled signal $\vec{\omega}_1$ is transformed into the roll-pitch-yaw angle rate vector, $\vec{\varphi}_1$, this was required to perform inverse kinematic analysis. The transformation is given by:

$$\vec{\varphi}_1 = {}^A T_B \vec{\omega}_1 \quad (19)$$

With ${}^A T_B$ a transformation commonly used to transform from angular velocity in coordinate frame \mathbf{B} to roll-pitch-yaw angle rates in coordinate frame \mathbf{A} , this transformation was shown previously²¹, and is given by:

$${}^A T_B = \begin{bmatrix} 1 & \text{sat}\beta & \text{cat}\beta \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha s\beta & c\alpha s\beta \end{bmatrix}$$

The signal $\vec{\varphi}_1$ is high-pass filtered to remove the sustained component of the roll-pitch-yaw angle rate and

perform washout as in the translational channel case. The signal $\vec{\varphi}_H$ is integrated to give the high frequency signal for roll-pitch-yaw angles ${}^A \vec{\varphi}_H$. The roll-pitch-yaw angle components for the sustained specific force vector \vec{f}_L and the vehicle angular velocity vector $\vec{\omega}_{veh}$ are combined to give the roll-pitch-yaw angle values used:

$${}^A \vec{\varphi} = {}^A \vec{\varphi}_L + {}^A \vec{\varphi}_H \quad (20)$$

Higher order filters are used due to more demanding accelerations experienced in a vehicle as compared to a flight simulator²². A 3rd order filter is used for HP Filter in the translational channel, 2nd order filter for LP Filter in the rotational channel and a 2nd order filter for HP Filter in the rotational channel.

4 Simulation System

Figure 5 illustrates the Matlab/Simulink simulation system containing all the components involved in the vehicle simulator data processing setup. This system includes input vehicle data scaling, classical washout algorithm and inverse kinematic implementations. The body sensor block below connects to the structural model and takes in the platforms position and orientation and presents it in a suitable form in the scope outputs. The simulation system designed was used to test the most important aspects in the vehicle simulator. These include testing:

- The inverse kinematics in replication of the trajectory of the input data.
- The classical washout algorithm in effectively performing platform washout.
- The effectiveness of the input data scaling and filter parameter selection, in the classical washout algorithm, in ensuring the motion platform constraints are not violated, this ensures safe operation on the actual 3-DOF motion platform.

The position control system, selected for the actuators, is a pneumatic system. It employs three pneumatically driven linear actuators and performs position control on each of these actuators. The system itself is limited to the following actuator position, velocity and acceleration values shown in table 1. Based on these constraints the vehicle input data scaling and classical washout filter parameters was varied. The objective was to get the actuator motion to be within the position, velocity and acceleration limits imposed.

Table 1: Actuator Motion Limits

Position	0.1 m
Velocity	0.2 m/s
Acceleration	2 m/s ²

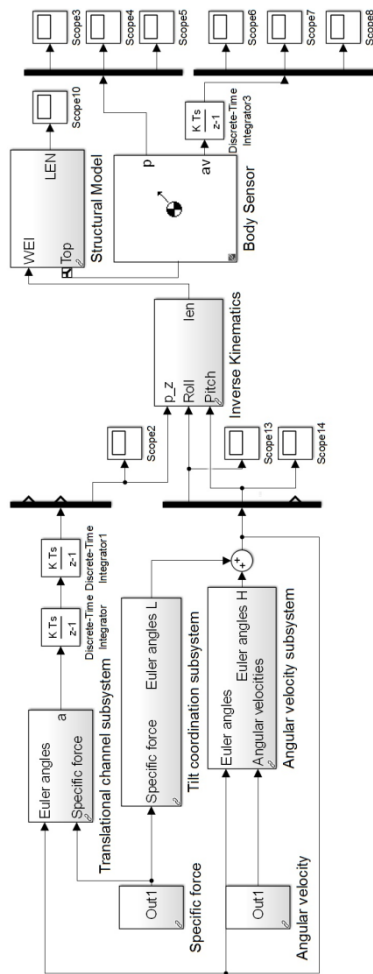


Figure 5: Vehicle Simulator Simulation System

5 Results and Discussion

The scaled input values selected in the classical washout algorithm for the simulation system in figure 5 are illustrated below in figures 6 and 7. The scaled specific force uses a scale factor of 0.5 to scale the input vehicle data. The scaled angular velocity uses a scale factor of 0.5 to scale the input vehicle data. These scale factors help ensure platform workspace limits are respected without significantly degrading the input vehicle data.

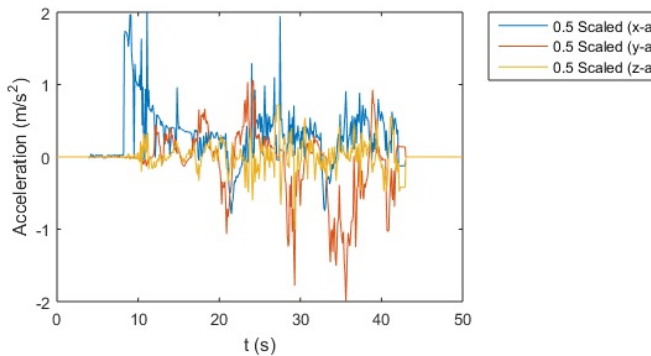


Figure 6: Scaled Specific Force Vehicle Data

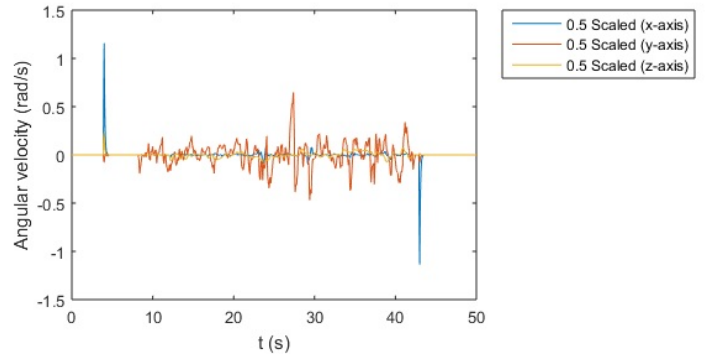


Figure 7: Scaled Angular Velocity Vehicle Data

The first part of the simulation verified the derived leg length equations. 11, 12 and 13, for the inverse kinematics, are correct and the washout process is effective in returning the platform to neutral (centre) position. The structural model block was built with SimMechanics and designed based on the geometrical structure of the vehicle simulator platform. The inverse kinematics block takes in the three independent motion parameters of roll, pitch and heave. These values are output from the classical washout algorithm. The inverse kinematics block outputs the actuator stroke lengths, which is based on the derived leg length equations 11, 12 and 13. The structural model will take in the actuator stroke lengths and move the actuators accordingly. The body sensor block outputs the platform trajectory, (position and orientation) to the scopes, for this actuator motion.

Figures 8 to 10 does a trajectory comparison for the three independent platform parameters of heave, roll and pitch. The input trajectory in these figures show the desired motion required to replicate the motion in the vehicle. The structural model output trajectory in these figures matches the input trajectory. The maximum errors are 0.0019 m for the heave (z-axis), 0.3074 deg for the roll (x-axis) and 0.2497° for the pitch (y-axis). The mean errors are 0.00036 m for the heave motion (z-axis), 0.0219° for the roll (x-axis) and 0.0289° for the pitch (y-axis). These error values are acceptable to be used in a vehicle simulator system, since these systems do not require high precision movements. These results show that the derived leg length equations 11, 12 and 13 are acceptable for the 3-DOF motion platform used in the vehicle simulator, since the input and output trajectories correlate. It can also be seen that the washout process is effective since the independent parameters of roll, pitch and heave all return to zero when the applied motion cue is complete. For the selected values for the input data scaling and filter parameters the maximum heave motion (z-axis) achieved for the platform was 0.0339 m and the maximum angle for the roll (x-axis) was 5.3° and pitch (y-axis) was 6.5°. These results represent reasonable performance for a research simulator.

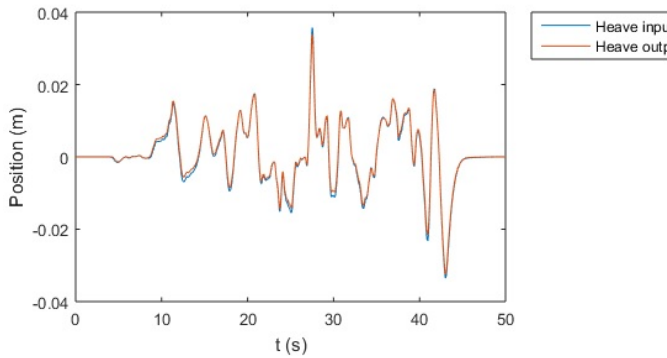


Figure 8: Heave Trajectory Comparison

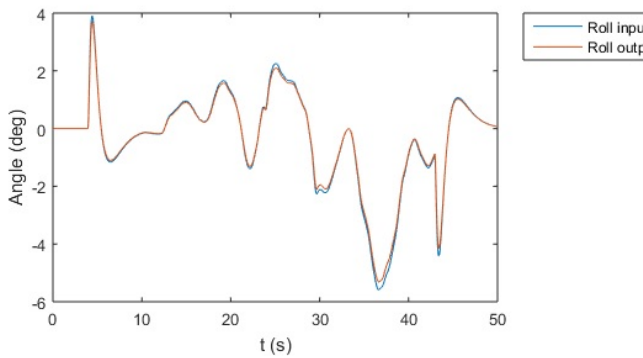


Figure 9: Roll Trajectory Comparison

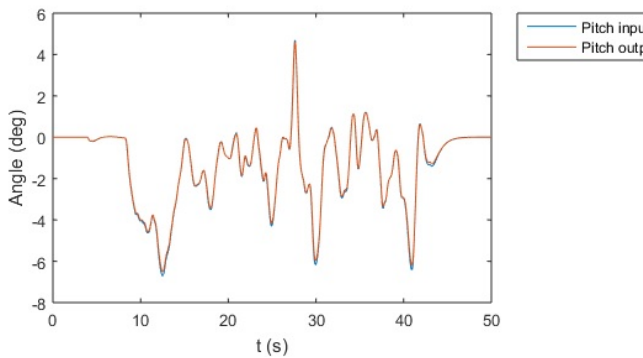
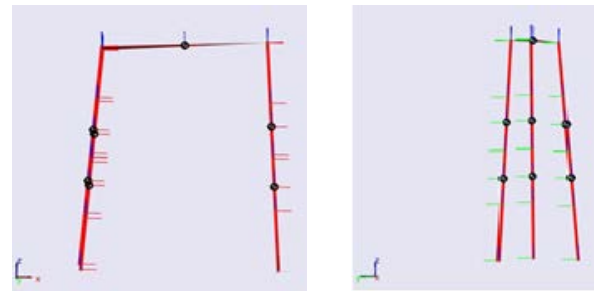


Figure 10: Pitch Trajectory Comparison

Figure 11 illustrates the 3-D model for the 3-DOF motion platform for two instances in the simulation; this system, using SimMechanics, provided a visualisation tool of expected platform motion. It was also used to verify platform motion is correct, for e.g. the platform tilts backward about the y-axis (pitch) to replicate a positive acceleration in the x-axis direction as in figure 11(a). In figure 11(b) the driver was going around a curve and turning left, therefore the driver was tilted to the right to replicate the motion sensations experienced in the vehicle.

The second and final aspect of the testing evaluated the actuator motion limits from the structural model output and determined if the actuator motions adhere to the constraints imposed. Figure 12 illustrates the actuator stroke length output from the structural model for each of the actuators. The platform sits at a height of 0.74 m at rest position, this means that $p_z = 0.74$ m at rest. At rest position the length of Actuator 1 is 0.7411 m, Actuator 2 is 0.7843 m and Actuator 3 is 0.7843 m. Actuator 1 had a maximum motion change of 0.0512 m, actuator 2 had a maximum motion

change of 0.0752 m and actuator 3 had a maximum motion change of 0.0729 m. From these results it is concluded that all three actuators adhere to the maximum possible actuator position limit of 0.1 m.



a) Platform Backward Tilt b) Platform Tilt to Right

Figure 11: Visual Representation of Platform Motion

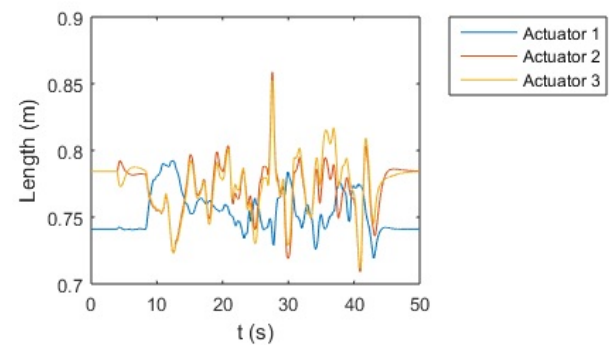


Figure 12: Structural Model Length Output of Actuators

Figure 13 illustrates the velocity output of each actuator from the structural model. Actuator 1 had a maximum velocity of 0.0937 m/s, actuator 2 had a maximum velocity of 0.1887 m/s and actuator 3 had a maximum velocity of 0.1897 m/s. The actuator velocity values all lie within the maximum permissible actuator velocity of 0.2 m/s.

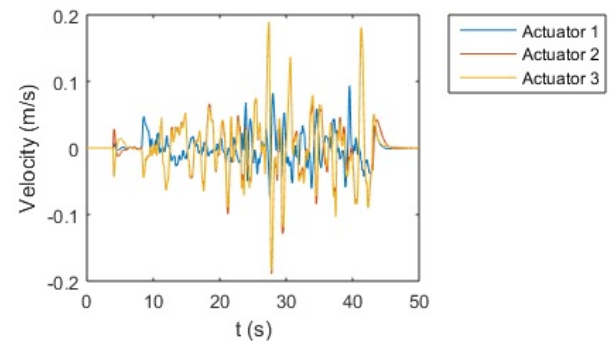


Figure 13: Structural Model Velocity Output of Actuators

Figure 14 illustrates the acceleration output of each actuator from the structural model. Actuator 1 had a maximum acceleration of 0.8076 m/s², actuator 2 had a maximum acceleration of 1.7322 m/s² and actuator 3 had a maximum acceleration of 1.7305 m/s². The accelerations of all actuators lie within the maximum acceleration value of 2 m/s².

The input data scaling and filter parameters selected ensured that none of the actuator motion constraints, in table 1, are violated. This result ensures safe operation for the driver in the vehicle simulator and safety of the

mechanical structure for the vehicle simulator. It can be seen from the acceleration of the actuator legs that the actuators reach maximum acceleration for very short periods in time. This is due to the fixed filter parameters of the classical washout algorithm which is designed for the worst motion case. Therefore platform workspace usage is not optimised for the classical washout algorithm.

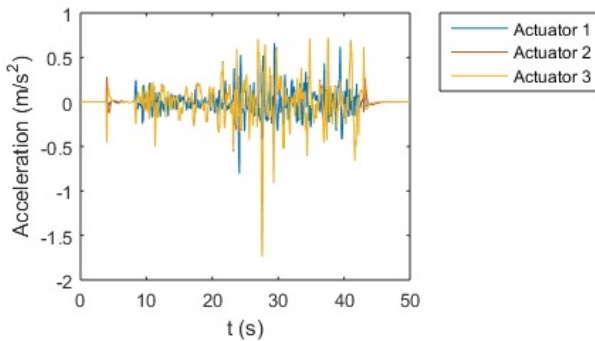


Figure 14: Structural Model Acceleration Output of Actuators

6 Conclusions and Future Work

This work presents a simulation system, developed in Matlab/Simulink, which allows for the vehicle data to be input directly into the system. Input data scaling, classical washout algorithm and inverse kinematics are performed in the system. The system facilitated for the appropriate input data scaling and filter parameters to be selected to ensure the actuator motion constraints are respected. The results show that the actuator motion constraints, in table 1, are respected. This ensures safe operation for the user and guarantees safety of the mechanical structure for the vehicle simulator. Inverse kinematic results obtained show that the derived inverse kinematics equations equations 11, 12 and 13 are correct and motion cueing results show effectiveness in performing washout.

The research contribution of this paper provides a simulation system to be used for a vehicle simulator. The simulation system assists in the design and development of the vehicle simulator, by ensuring the input data scaling and filter parameters are selected to ensure safety of the user and the vehicle simulator. Results from this work are applicable to all commercial vehicle simulators which make use of a parallel manipulator motion platform. In future work the addition of improved motion cueing techniques will be implemented. This will facilitate improved workspace utilisation and position control performance.

Acknowledgements

This work is funded by the Council for Scientific and Industrial Research (CSIR) in South Africa.

References

1. Patel YD and George PM, Parallel Manipulators Applications - A Survey, *Modern Mechanical Engineering*, 2012, 2 (3), 57-64.
2. Gough VE and Whitehall SG, Universal Tyre Test Machine, *Proceedings of 9th International Congress FISITA*, Institution of Mechanical Engineers, London, 1962, 117-135.

3. Stewart D, A Platform with Six Degrees of Freedom, *Proceedings of the Institution of Mechanical Engineers*, 1965, 180 (1), 371-386.
4. Tsai LW, Walsh GC and Stamper RE, Kinematics of a Novel Three DOF Translational Platform, *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, IEEE, Minneapolis, Minnesota, 1996, 3446-3451.
5. Lee K and Shah DK, Kinematic Analysis of a Three Degrees of Freedom In-Parallel Actuated Manipulator, *IEEE Journal of Robotics and Automation*, 1988, 345-360.
6. Drosdol J and Panik F, The Daimler-Benz Driving Simulator A Tool for Vehicle Development, *SAE Paper 850334*, 1985, SAE.
7. Bertollini GP, Johnston CM, Kuiper JW, Kukula JC, Kulczycka, MA and Thomas, WE, The General Motors Driving Simulator, *SAE Technical Paper 940179*, 1994, SAE.
8. Freeman JS, Watson G, Papelis YE, Lin TC, Tayyab A, Romano RA and Kuhl JG, The Iowa Driving Simulator: An Implementation and Application Overview, *SAE Technical Paper 950174*, 1995, SAE.
9. TOYOTA MOTOR CORPORATION, Toyota Develops World-class Driving Simulator, http://www.toyota.co.jp/en/news/07/1126_1.html.
10. Kemeny A and Panerai F, Evaluating perception in driving simulator experiments, *Trends in Cognitive Sciences*, 2003, 7 (1), 31-37.
11. Capustiac A, Banabic D, Schramm D and Ossendoth U, Motion Cueing: From Design until Implementation, *Proceedings of the Romanian Academy, Series A*, 2011, 12 (3), 249-256.
12. Schmidt SF and Conrad B, Motion Drive Signals for Piloted Flight Simulators, 1970, Contractor Report NASA CR-1601, NASA.
13. Ariel D and Sivan R, False Cue Reduction in Moving Flight Simulators, *IEEE Transactions on Systems, Man and Cybernetics*, 1984, 14 (4), 665-671.
14. Sivan R, Ish-shalom J and Huang J, An Optimal Control Approach to the Design of Moving Flight Simulators, *IEEE Transactions on Systems, Man and Cybernetics*, 1982, 12 (6), 818-827.
15. Baseggio M, Beghi A, Bruschetta M, Maran F and Minen D, An MPC Approach to the Design of Motion Cueing Algorithms for Driving Simulators, *Intelligent Transportation Systems (ITSC)*, 14th International IEEE Conference on Intelligent Transportation Systems, Washington, D.C., 2011, 692-697.
16. Strayer DL and Drews FA, Simulator Training Improves Driver Efficiency: Transfer from the Simulator to the Real World, *Proceedings of the Second International Driving Symposium on Human Factors in Driver Assessment Training and Vehicle Design*, Park City, UT, 2005, 190-193.
17. Kihl M and Wolf PJ, Using Driving Simulators to Train Snowplow Operators: The Arizona Experience, *Proceedings of the 2007 Mid-Continent Transportation Research Symposium*, Ames, Iowa, 2007, 16-17.
18. SimMechanics User's Guide, The MathWorks, Inc., http://www.mathworks.com/help/pdf_doc/physmod/sm/sm Ug.pdf.

19. Yu L, Zhang L, Zhang N, Yang S and Wang D, Kinematics Simulation and Analysis of a 3-RPS parallel robot on SimMechanics, *Proceedings of the 2010 IEEE International Conference on Information and Automation*, Harbin, China, 2010, 2363-2367.
20. Shiong CY, Jalil MK and Hussein M, Motion Visualisation and Control of a Driving Simulator Motion Platform, *Proceeding of the 6th International Symposium on Mechatronics and its Applications (ISMA09)*, Sharjah, UAE, 2009, 1-5.
21. Reid LD and Nahon MA, Flight Simulation Motion-Base Drive Algorithms: Part 1, Developing and Testing the Equations, UTIAS Report No. 296, University of Toronto Institute for Aerospace Studies, 1985.
22. Reid LD and Nahon MA, Flight Simulation Motion-Base Drive Algorithms: Part 2, Selecting the System Parameters, UTIAS Report No. 307, University of Toronto Institute for Aerospace Studies, 1986.