Actuator State Based Adaptive Motion Drive Algorithm

Peter Grant and Amir Naseri

University of Toronto Institute for Aerospace Studies,

4925 Dufferin Rd, Toronto, Ontario, Canada,

Ph. 416-667-7726,

E-mail: prgrant@utias.utoronto.ca

Abstract

A new Actuator State Based Adaptive (ASBA) motion drive algorithm was developed. In contrast to classical motion drive algorithms a subset of the motion drive parameters are time varying. The ASBA algorithm uses the steepest descent algorithm to adapt these parameters to minimize a pre-defined cost function. The cost function contains penalties on motion cue errors, simulator motion, and the distance of the adaptive parameters from their nominal values. The newly developed adaptive algorithm uses actuator states in the cost function rather than the traditional inertial Cartesian motion states. The ASBA algorithm is therefore able to adapt more "intelligently" to driving maneuvers that generate motion in multiple (Cartesian) degrees-of-freedom (such as a turn plus braking maneuver). The new adaptive motion drive algorithm also uses a variable step size in the steepest descent algorithm. The step size is a function of simulator motion: large amounts of motion lead to a reduction in the steepest descent step size. This reduction in step size eliminates a previously documented adaptive algorithm instability.

Nomenclature

K_f - specific force gain
K_ω - angular rate gain
s - Laplace operator
a - acceleration vector
g - gravity vector
f_V - specific force at vehicle reference point
ω_V - vehicle angular velocity
β - vector of Euler angles $[\phi, \theta, \psi]^T$
β_{1LP} -unfiltered tilt-coordination Euler angles
β_{1HP} -high-pass channel unfiltered Euler angles
β_{LP} - low-passed tilt-coordination Euler angles
β_{HP} - high-passed simulator Euler angles
β_S - total simulator Euler angles
L_{IS} -simulator body frame to inertial frame transformation matrix
T_S - simulator body angular velocity to Euler rate transformation matrix
a_2 - unfiltered inertial acceleration
a_C - simulator (filtered) inertial acceleration
S_I - simulator inertial displacement
$\omega_{hpx,y,z}$ - x,y and z translational 2nd-order high-pass filter break freq.
$\varsigma_{hpx,y,z}$ - x,y,and z translational 2nd-order high-pass damping
$\omega_{bx,y,z}$ - x,y and z translational 1st-order high-pass break freq.
$\omega_{hp\phi,\theta,\psi}$ - roll, pitch and yaw rotational 2nd-order high-pass filter break freq.
$\varsigma_{hp\phi,\theta,\psi}$ - roll, pitch and yaw rotational 2nd-order high-pass damping
$\omega_{lpx,y}$ - x and y tilt-coordination 2nd-order low-pass break freq.
$\varsigma_{lpx,y}$ - x and y tilt-coordination 2nd-order low-pass damping
$P_{x,y,z}$ -x,y and z adaptive gains
$P_{\phi,\theta,\psi}$ - roll, pitch and yaw adaptive gains
$P_{x0,y0,z0}$ -x,y and z adaptive gain nominal values
$P_{\phi 0,\theta 0,\psi 0}$ - roll, pitch and yaw adaptive gain nominal values
W_{ij} -cost function weights
$J_{x,y,z,\psi}$ -cost function
$G_{x,y,z,\phi,\theta,\psi}$ -steepest descent step size
$G_{x0,y0,z0,\phi 0,\theta 0,\psi 0}$ -steepest descent nominal step
L_i -ith-actuator extension
$\dot{X}$ -first time derivative of X

Introduction

High-fidelity motion-based simulators often employ Stewart platform motion systems to provide drivers with the appropriate vestibular, proprioceptive, and tactile motion cues while driving. Since most driving maneuvers involve motions that greatly exceed a simulator's motion displacement capability, the calculated vehicle motions must be filtered before they are sent to the motion system. This filtering is implemented by the Motion Drive Algorithm (MDA). The MDA is designed and tuned to minimize errors between the calculated vehicle and simulator motions while keeping the simulator within

its motion envelope. There are a number of different algorithm designs currently used within the driving simulator community. One of the most common algorithm designs is the classical algorithm, which uses a combination of high-pass filters, low-pass filters, scale factors, limiters and coordinate transformations to filter the vehicle motion. The filter parameters within the classical algorithm are time invariant, although they are often switched between distinct sets of parameters for particular driving maneuvers. The classical MDA parameters are tuned prior to the driving task and are fixed for each segment of the drive. In contrast, an adaptive algorithm (first developed by Parrish et al [1]), continually adjusts the MDA parameters in real-time based on simulator and vehicle states. Previous motion studies in flight simulation have shown that adaptation is a useful feature [1,2]. The adaptive algorithm allows unexpectedly large inputs (such as heavy braking) to be more heavily filtered and thereby avoiding limiting algorithms, while more modest inputs (such as lane tracking and speed maintenance) are lightly filtered, thereby providing useable cues to the driver. Although the adaptive MDA has shown promising results, it is still not widely used for motion-base simulators because it does not show large improvements over the classical algorithm, it is prone to instability, and it has a more complicated tuning procedure. The new actuator state based adaptive motion drive algorithm attempts to address these problems, particularly when a Stewart platform is employed.

Adaptive MDAs (such as the UTIAS adaptive MDA [3], referred to herein simply as the UTIAS algorithm) adapt their parameters by minimizing a cost function in real-time. The cost function has penalties on simulator motion states, motion cueing errors and distance of the adaptive parameters from their nominal values. The simulator motion states in the cost functions are typically expressed in a Cartesian inertial frame. The parallel architecture of the Stewart platform leads to a strong coupling between the maximum motion on one Cartesian degree-of-freedom and the amount of motion on the other five degrees-of-freedom. The motion workspace is therefore a complex function of all six Cartesian degrees-of-freedom. In actuator space however, the workspace is completely uncoupled; the six actuator limits are independent of each other. A new Actuator State Based Adaptive algorithm (ASBA) was developed to take advantage of this by using motion system actuator states in cost function instead of the Cartesian based motion states. The ASBA algorithm works similar to the UTIAS algorithm in cases for single degree-of-freedom maneuvers. For more complex combined degree-of-freedom maneuvers however, the ASBA avoids actuator limiting in cases where the UTIAS algorithm results in one or more actuators limiting.

The UTIAS algorithm is prone to very lightly damped oscillatory response for large inputs [4]. This lightly damped oscillatory response leads to large false cues and the associated large reduction in motion fidelity. This problem was overcome in the current study by making the steepest descent step size a function of the simulator motion rather than a constant. The benefits of the adaptive steepest descent step size are demonstrated by comparing the response of the UTIAS algorithm with and without the adaptive step size to a step input in Z specific force. Moreover, the benefits of the new ASBA algorithm are demonstrated by comparing the response of the ASBA algorithm to the UTIAS algorithm to a combined turning and braking maneuver.

The UTIAS Adaptive Motion Drive Algorithm

The UTIAS Adaptive MDA [3] is closely based on the classical motion drive algorithm developed by Reid and Nahon [2] and the original adaptive algorithm developed by Parrish et al [1]. The UTIAS adaptive algorithm is identical to the classical algorithm except for the adaptation of the high-pass filter parameters. Figure 1 below shows the block diagram of the UTIAS adaptive algorithm.

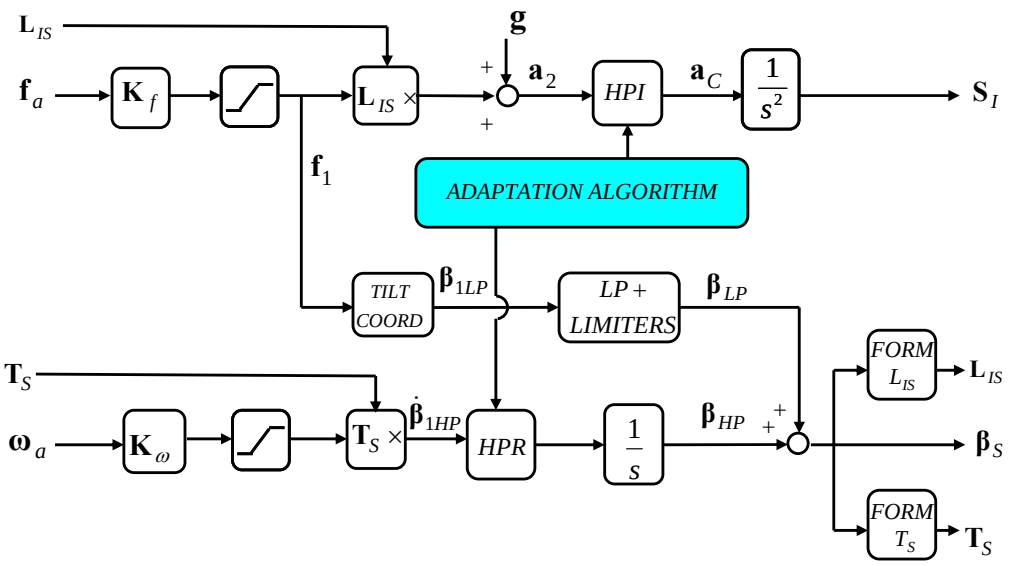

Figure 1 UTIAS Adaptive Algorithm

The HPI block contains three filters, one for the X, Y and Z inertial accelerations. The HPR block also contains three filters, one for roll, pitch and yaw Euler angle rotations. In this study only gain adaptation is considered. Kirdeikis [5] previously showed that these are the most advantageous parameters to adapt. The HPI filters are 3rd order high-pass filters. In the Laplace domain, they are given as,

$$\frac{a_c^{x,y,z}}{P_{x,y,z}\,a_2^{x,y,z}} = \frac{s^2}{s^2 + 2\varsigma_{hpx,y,z}\,\omega_{hpx,y,z}\,s + \omega_{hpx,y,z}^2}\,\frac{s}{s+\omega_{bx,y,z}}$$

The HPR filters are either first or second order high-pass filters given as either,

$$2nd-order\ \frac{\dot{\beta}_{HP}^{\phi,\theta,\psi}}{P_{\phi,\theta,\psi}\,\dot{\beta}_{1HP}^{\phi,\theta,\psi}} = \frac{s^2}{s^2 + 2\varsigma_{hp\phi,\theta,\psi}\,\omega_{hp\phi,\theta,\psi}\,s + \omega_{hp\phi,\theta,\psi}^2} \quad 1st-order\ \frac{\dot{\beta}_{HP}^{\phi,\theta,\psi}}{P_{\phi,\theta,\psi}\,\dot{\beta}_{1HP}^{\phi,\theta,\psi}} = \frac{s}{s+\omega_{hp\phi,\theta,\psi}}$$

The filter order can be selected independently for each Euler angle filter. Cross-feed between the specific force and rotational channels provides low-frequency tilting of the g-vector in the body frame of the driver. This tilting of the g-vector in turn provides low-frequency, sustained horizontal plane specific force cues to the driver. Neglecting tilt rate and acceleration limiting, the low-pass filters are given by,

$$\frac{\beta_{LP}^{\theta,\phi}}{\beta_{1LP}^{x,y}} = \frac{\omega_{lpx,y}{}^2}{s^2 + 2\varsigma_{lpx,y}\,\omega_{lpx,y}\,s + \omega_{lpx,y}^2}$$

The adaptive gains in these transfer functions, P_x,P_y,P_z,P_ϕ,P_θ,P_ψ , are then found by minimizing the following four cost functions:

X/Pitch:

$$J_x = \frac{1}{2}\Big[W_{x0}(a_2^x - \ddot{S}_I^x)^2 + W_{x1}(\dot{\theta}_{1HP} - \dot{\theta}_S)^2 + W_{x2}(\dot{S}_I^x)^2 + W_{x3}(S_I^x)^2 + W_{x4}(\dot{\theta}_S)^2 + W_{x5}(\theta_S)^2 + W_{x6}(P_x - P_{x0})^2 + W_{x7}(P_\theta - P_{\theta 0})^2\Big]$$

Y/Roll:

$$J_y = \frac{1}{2}\Big[W_{y0}(a_2^y - \ddot{S}_I^y)^2 + W_{y1}(\dot{\phi}_{1HP} - \dot{\phi}_S)^2 + W_{y2}(\dot{S}_I^y)^2 + W_{y3}(S_I^y)^2 + W_{y4}(\dot{\phi}_S)^2 + W_{y5}(\phi_S)^2 + W_{y6}(P_y - P_{y0})^2 + W_{y7}(P_\phi - P_{\phi 0})^2\Big]$$

Heave (Z):

$$J_z = \frac{1}{2}\Big[W_{z0}(a_2^z - \ddot{S}_I^z)^2 + W_{z1}(\dot{S}_I^z)^2 + W_{z2}(S_I^z)^2 + W_{z3}(P_z - P_{z0})^2\Big]$$

Yaw:

$$J_\psi = \frac{1}{2}\Big[W_{\psi 0}(\dot{\psi}_{1HP} - \dot{\psi}_S)^2 + W_{\psi 1}(\dot{\psi}_S)^2 + W_{\psi 2}(\psi_S)^2 + W_{\psi 3}(P_\psi - P_{\psi 0})^2\Big]$$

For the X/Pitch and Y/Roll cost functions, the first two terms are penalties on motion cue errors, the next four terms are penalties on the simulator motion and the last two terms are penalties on the distance of the adaptive parameters from their nominal values. For the heave and yaw cost functions the first term is a motion cue error penalty, the second and third terms are penalties on simulator motion and the last term is a penalty on the distance of the adaptive parameter from its nominal value. The W_{ij} are weights on each term in the cost function. They are selected to achieve the desired filter response.

The steepest descent method is used to continuously minimize the above cost functions. The time derivative of the adaptive parameters are determined at each time step of the simulation using,

$$\dot{P}_x = -G_x \frac{\partial J_x}{\partial P_x},\ \dot{P}_y = -G_y \frac{\partial J_y}{\partial P_y},\ \dot{P}_z = -G_z \frac{\partial J_z}{\partial P_z},\ \dot{P}_\theta = -G_\theta \frac{\partial J_x}{\partial P_\theta},\ \dot{P}_\phi = -G_\phi \frac{\partial J_y}{\partial P_\phi},\ \dot{P}_\psi = -G_\psi \frac{\partial J_\psi}{\partial P_\psi}$$

Where the $G_{x,y,z,\phi,\theta,\psi}$ are the steepest descent step sizes. The adaptive algorithm is tuned for a particular maneuver by the selection of the classical high-pass parameters, the weights on the individual terms in the cost function, and the steepest descent step sizes.

Adaptive Steepest Descent Step Size

The first improvement made to the UTIAS adaptive algorithm was the addition of an adaptive steepest descent step size. This is a common technique in adaptive filter theory. The current UTIAS adaptive algorithm uses a fixed steepest descent step size to project the solution for the adaptive parameters in the direction of the steepest descent. If the step size is selected to achieve reasonable adaptation when the cost function has a modest slope, then a steeply sloped cost function leads to oscillatory behavior. The new adaptive algorithm adapts the step size based on the following equations,

$$G_{x,y,z} = \frac{G_{x0,y0,z0}}{\left(1+\left|\dot{S}_I^{x,y,z}\right|+\left|S_I^{x,y,z}\right|\right)^N} \text{ or } G_{\phi,\theta,\psi} = \frac{G_{\phi 0,\theta 0,\psi 0}}{\left(1+\left|\dot{\beta}_S^{\phi,\theta,\psi}\right|+\left|\beta_S^{\phi,\theta,\psi}\right|\right)^N}$$

Where β_S is a vector of the three standard Euler angles. N is the order of the step size adaptation, and it was 2 in the current study. These equations reduce the step size for large simulator motions. Since there is a strong relationship between the slope of the cost function and the amount of simulator motion, these equations reduce the steepest descent step size when the cost function is steep.

Actuator State Based Adaptive Motion Drive Algorithm

The block diagram for the ASBA motion drive algorithm is shown in Figure 1; it is identical to the UTIAS adaptive algorithm. The high-pass and low-pass filters blocks are identical to those listed previously for the UTIAS adaptive algorithm. The difference between the two algorithms is completely contained within the ADAPTATION ALGORITHM block. The cost functions for the actuator state based adaptive algorithm are:

X/Pitch:

$$J_x = \frac{1}{2}\left[W_{x0}(a_2^x - \ddot{S}_I^x)^2 + W_{x1}(\dot{\theta}_{1HP} - \dot{\theta}_S)^2 + W_{x2}(\sum_{i=1`}^{6} L_i^4) + W_{x3}((\sum_{i=1`}^{6} \dot{L}_i^2) + + W_{x4}(P_x - P_{x0})^2 + W_{x5}(P_\theta - P_{\theta 0})^2\right]$$

Y/Roll:

$$J_y = \frac{1}{2}\left[W_{y0}(a_2^y - \ddot{S}_I^y)^2 + W_{y1}(\dot{\phi}_{1HP} - \dot{\phi}_S)^2 + W_{y2}(\sum_{i=1`}^{6} L_i^4) + W_{y3}((\sum_{i=1`}^{6} \dot{L}_i^2) + + W_{y4}(P_y - P_{y0})^2 + W_{y5}(P_\phi - P_{\phi 0})^2\right]$$

Heave (Z):

$$J_z = \frac{1}{2}\left[W_{z0}(a_2^z - \ddot{S}_I^z)^N + W_{z1}(\sum_{i=1}^{6} L_i^4) + W_{z2}(\sum_{i=1}^{6} \dot{L}_i^2) \quad + W_{z3}(P_z - P_{z0})^2\right]$$

Yaw:

$$J_\psi = \frac{1}{2}\left[W_{\psi 0}(\dot{\psi}_{1HP} - \dot{\psi}_S)^2 + W_{\psi 1}(\sum_{i=1}^{6} L_i^4) + W_{\psi 2}(\sum_{i=1}^{6} \dot{L}_i^2) + W_{\psi 3}(P_\psi - P_{\psi 0})^2\right]$$

The penalty terms for inertial Cartesian displacements and velocities have been replaced by penalty terms on the actuator lengths and velocities. The actuator penalty is raised to the power of 4 to compensate for the fact that some motion may result in significant displacement of a small subset of the actuators. As with the UTIAS algorithm the steepest descent algorithm is used to move towards the cost function minimum at each time step.

Results

Adaptive Steepest Descent Step Size

The benefits of the variable step size are demonstrated by examining the response of the algorithm to a small and large step in specific force. To simplify the analysis an input in Z-specific force is considered. The UTIAS algorithm with and without the variable step size were identically tuned for a 1m/s^2 step input. As shown in Figure 2, the responses of the UTIAS algorithm with and without the variable step size are almost identical.

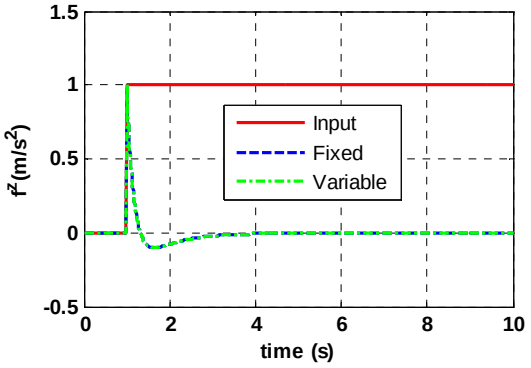

Figure 2 Z-Specific Force Response, 1m/s^2 Step

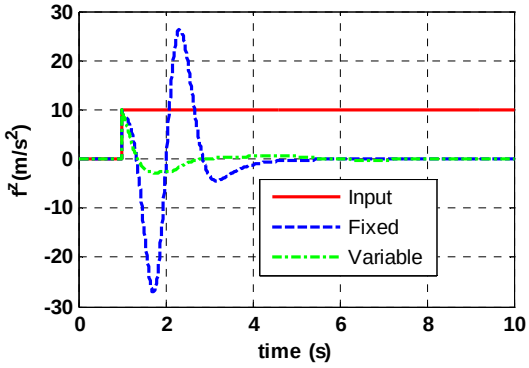

Figure 3 Z-Specific Force Response, 10 m/s^2 Step

The response of the UTIAS algorithm with and without the variable step size (and the same set of parameters as for the 1m/s^2 case) to a step input in Z-specific force of 10 m/s^2 are shown in Figure 3. As can be seen in this figure, the algorithm with the fixed step size results in large oscillations in specific force that overshoot the initial input. These oscillations would likely lead to serious false cue complaints. The response of the UTIAS algorithm with the variable step size has no such oscillation; in fact, it looks very similar to a high-pass filter response. Figures 4 and 5 show the cost (J_z) and the adaptive gain P_z, versus time for the 10 m/s^2 input. From these figures, it can be seen that the cost function minimum is being overshot when the fixed steepest descent step size is used. The initial response is quite similar between the fixed and variable step size when the simulator motion is modest (1s to 1.3s) but as the simulator motion builds and the cost function grows steeper, the steepest descent algorithm projects too far in the direction of steepest descent.

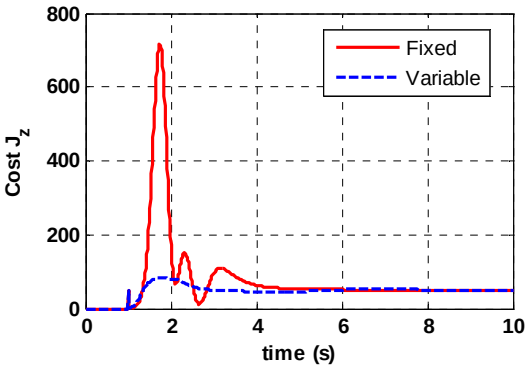

Figure 4 Cost Function Step Input of 10 m/s^2

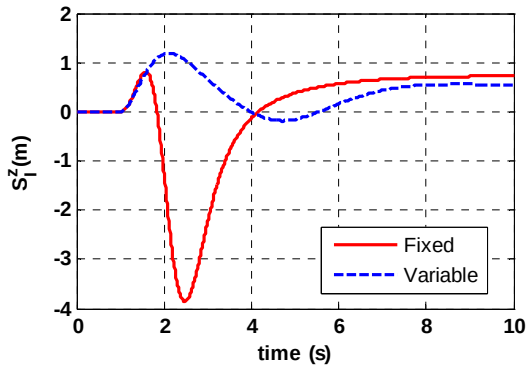

Figure 5 Adaptive Z-Gain Step Input of 10 m/s^2

Actuator State Based Adaptation

The advantage of the actuator state based adaptation will be demonstrated by comparing the response of the UTIAS and the ASBA algorithms for three maneuvers: a modest turn at 72 kph, a modest braking maneuver from 72 kph, and a combined turn and brake maneuver from 72 kph. These maneuvers are representative of a serpentine rural roadway with gentle increases in curvature at turn entry and their frequency content is particular troublesome for typical Stewart platform motion systems (the maneuver requires a large amount of simulator displacement for high-fidelity simulation). The two algorithms will be tuned such that they give similar results for the turn and brake maneuvers, with a reasonable amount of adaptation to prevent actuator limiting. The same tuning will then be used for the combined braking and turning maneuver. For the simulations a Stewart platform with an actuator stroke of 1625.6mm was used, and a nominal gain of 0.4 was used for the X and Y specific forces.

Turn Maneuver

The lateral specific force response of the UTIAS, the ASBA and the underlying classical algorithm (obtained by setting the steepest descent step size to zero) are shown in Figure 6 along with the vehicle specific force. The vehicle specific force is scaled by the overall gain so the shapes of the specific force responses can be compared.

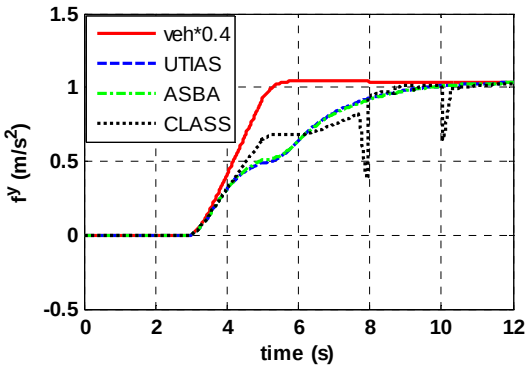

Figure 6 Y Specific Force Response for Turn

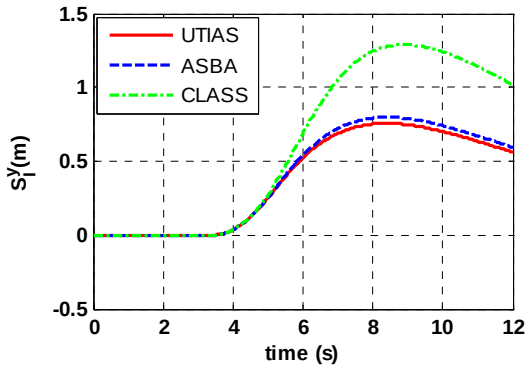

Figure 7 Y Displacement for Turn (Ignoring Simulator Limits)

The UTIAS and ASBA specific force responses are almost identical; they follow the vehicle input reasonably well for 1.5 seconds and then asymptote for approximately 1 second before the tilt-coordination eventually builds up to provide accurate steady-state response. The classical algorithm response is identical for the first 1.5 seconds of the turn entry but then the adaptive algorithms start to scale back the response of the simulator relative to the classical. The classical algorithm eventually leads to actuator limiting at 8 seconds resulting in a large false cue. If we examine the simulator lateral displacement (ignoring the limitations of the Stewart platform) in Figure 7, we can see that the adaptation reduces the displacement by about 30% for both adaptive algorithms.

Brake Maneuver

The X-specific force and X-displacement for the UTIAS, ASBA, and the classical algorithms are shown in Figures 8 and 9 (the limitations of the Stewart platform were not considered for the displacement responses). The specific force responses of the two adaptive algorithms are almost identical and significant adaptation is required to prevent actuator limiting.

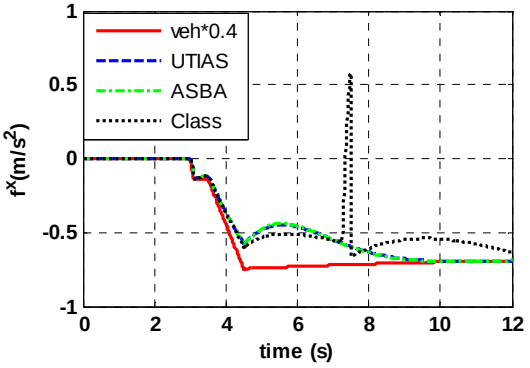

Figure 8 X-Specific Force for Braking

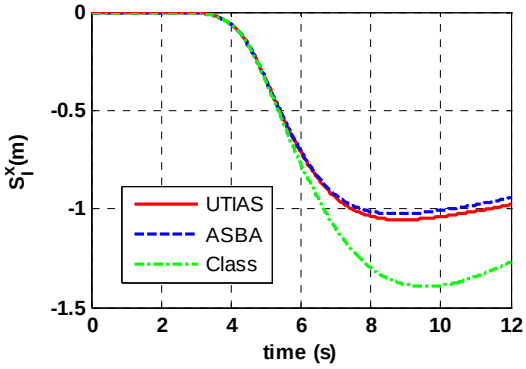

Figure 9 X-Displacement for Braking

Combined Turn and Brake

For the combination of the turn and braking maneuvers the tuning of the two algorithms was kept identical to the individual brake and turn maneuvers. The specific force responses of the two algorithms are shown in Figures 10 and 11.

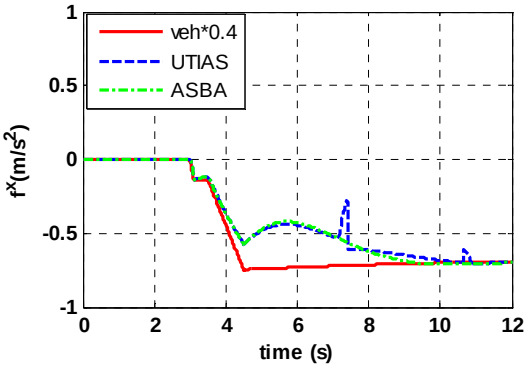

Figure 10 X-Specific Force Combined Maneuver

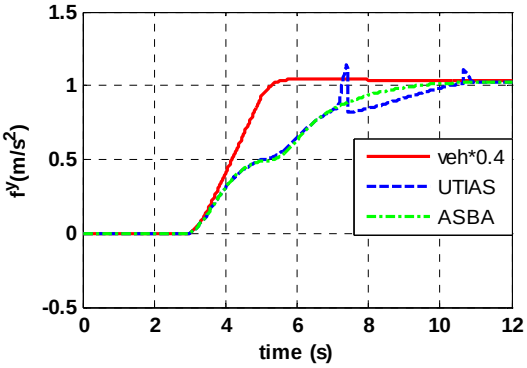

Figure 11 Y-Specific Force Combined Maneuver

The UTIAS adaptive algorithm response has actuator limiting occurring at 7 seconds whereas the new ASBA adapts to the combined maneuver and does not lead to actuator limiting. If we examine Figure 12, we see that the unlimited X and Y simulator displacement time histories for the UTIAS algorithm are almost identical to the respective displacements for the individual turn and braking maneuvers. Since the UTIAS algorithm cost function uses the Cartesian displacement (and derivative), the adaptation of P_x and P_y is therefore almost identical to the individual maneuvers as seen in Figures 14 and 15. Due to the parallel architecture of the Stewart platform, the combination of the

brake and turn maneuvers leads to increased displacement of some of the actuators relative to either individual maneuver. As shown in Figure 13, actuator 5 has an increased displacement. Since the ASBA cost function includes the length of the actuators there is increased adaptation of both P_x and P_y compared to the individual maneuvers, see Figures 14 and 15. This increased adaptation prevents the limiting that occurs for the UTIAS algorithm.

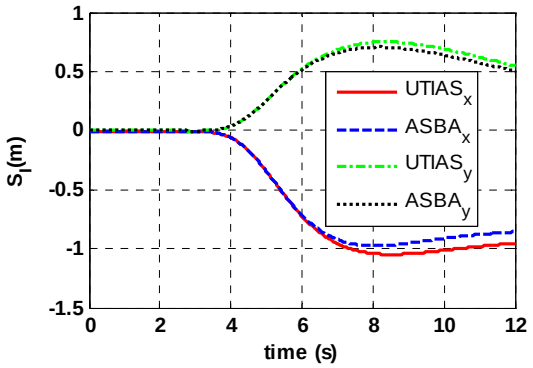

Figure 12 Displacement, Combined Maneuver

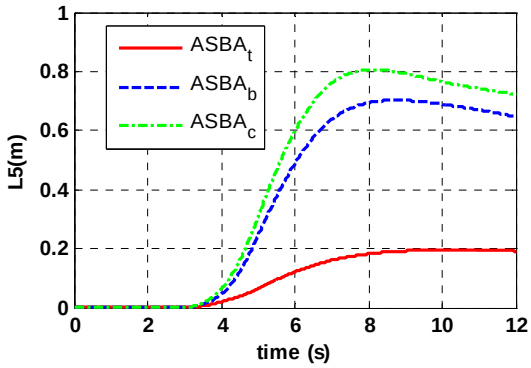

Figure 13 Actuator 5, Combined Maneuver

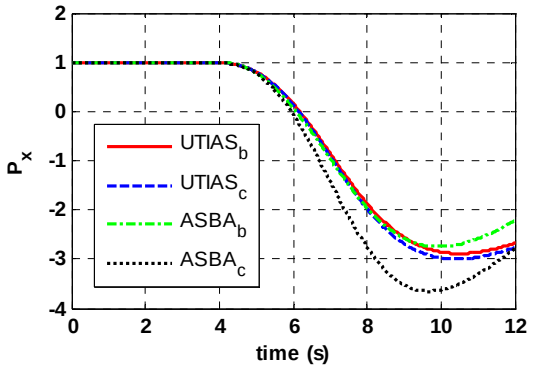

Figure 14 P_x Brake and Combined Maneuver

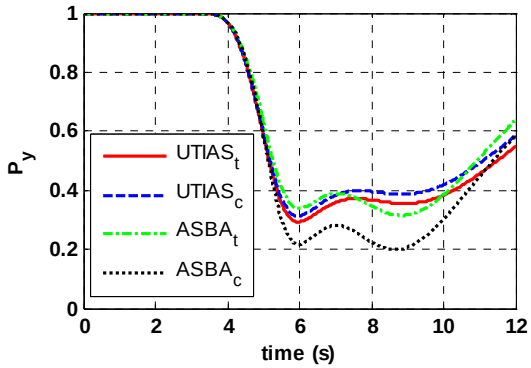

Figure 15 P_y Turn and Combined Maneuver

Conclusions

A new actuator state based adaptive motion drive algorithm that includes a variable steepest descent step size was developed. The new ASBA algorithm has improved stability and adaptation compared to the previous UTIAS adaptive algorithm. Further improvements in the algorithm are expected by replacing the four cost functions by a single cost function and by the inclusion of a penalty based on an estimate of the time for each actuator to reach a displacement limit. These are the subject of future investigation.

1 Parrish, R.V., Dieudonne, J. E. and Bowles, R. L., Coordinated Adaptive Washout for Motion Simulators, AIAA Journal of Aircraft, Vol. 12, No. 1, Jan., 1975.
2 Reid, L. D. and Nahon M. A., Flight Simulation Motion-Base Drive Algorithms: Part 1-Developing and Testing the Equations, UTIAS Report 296, Toronto, 1985.
3 Nahon, M. A., Reid L. D., and Kirdeikis, J., Adaptive Simulator Motion Software with Supervisory Control, Journal of Guidance, Control and Dynamics, Vol. 15, No. 2, pp 376-383, 1992.
4 O'Toole, G., Investigation of the UTIAS Adaptive Algorithm, M. A. Sc. Thesis, UTIAS, 1995.
5 Kirdeikis J., Evaluation of Nonlinear Motion-Drive Algorithms for Fight Simulators, UTIAS Technical Note 272, UTIAS, June 1989.