HYBRID MOTION CUEING ALGORITHMS FOR REDUNDANT ADVANCED DRIVING SIMULATORS

Hatem Elloumi, Nadia Maïzi and Marc Bordier
Centre de Mathématiques Appliquées, École des Mines de Paris
2004 route des Lucioles, 06902 Sophia Antipolis Cedex, France

Keywords: Driving simulation, motion cueing, hybrid control.

Abstract: Redundant Advanced Driving Simulators (hexapods mounted on rails) present an extra capability to reproduce motion sensations. The exploitation of this capability is currently done by frequency separation methods without taking into account the frequency overlapping between the hexapod and the rails. Within this bandwidth, these two degrees of freedom could be considered as equivalent. Our aim is to use this equivalence to improve the motion restitution. We offer two algorithms based on the hybrid systems framework which deal with the longitudinal mode. Their goal is to improve the restitution of motion sensations by reducing false cues (generated by actuators braking) and decreasing null cues (due to actuators blocking). Our algorithms include and treat all steps of motion cueing: motion tracking (restitution), braking before reaching the displacement limits, washout motion, and switching rules.

1 INTRODUCTION

Driving simulators are advanced devices composed of four components: a virtual scene projected on a wide screen to imitate the road and the traffic, an audio system to play the driving sounds (horn, squeal of brakes, etc.), a car cockpit (including a real dashboard, the pedals and the seat of the driver) to copy the body position and the interaction of the driver with a real vehicle and finally a robot carrying the car cockpit to provide its motion. While the first three components could be considered as offering a sufficiently high degree of realism, the robot presents a very low capacity of displacement, thus preventing it from performing the real car motions.

In fact, the aim of a driving simulator is not tracking real trajectories produced by outdoors driving but reproducing the corresponding motion *sensations*. How could we then, generate realistic motion *sensations* in simulation despite the constrained robot motion? It is the aim of Motion Cueing Algorithms (MCA) to give heuristically an answer to this problem. This paper uses the hybrid systems framework (Zaytoon, 2001; Van der Shaft and Schumacher, 1999) to build two MCA designed specifically for redundant simulation robots. These robots are made up of two parts:

- A Gough-Stewart parallel robot or hexapod (this parallel robot is composed of three parts: a moving body called the *platfom* (carrying the car cockpit) linked to the *base* through six extensible legs. Each leg is composed of a prismatic joint (i.e. an electro hydraulic jack) and two passive spherical joints making the connection with the base and the platform. For an excellent overview of parallel robots the reader is referred (Merlet, 2000)). The jacks' excursions are of ± 20cm allowing a six-dimension motion of the car cockpit up to ± 15cm (in linear directions) and up to 30deg (for rotations).
- A rail system carrying the base of the hexapod to provide an extra motion in the horizontal plane. In this paper the rails' limits are: ± 2m.

(Elloumi, 2006) shows that rails and the hexapod present overlapping bandwidths in the high frequency domain. So how could we benefit from this redundancy? Our approach is based on the *classical* MCA which will be presented in section 3. But before addressing this point, section 2 will deal with the notion

Elloumi H., Maïzi N. and Bordier M. (2007).
HYBRID MOTION CUEING ALGORITHMS FOR REDUNDANT ADVANCED DRIVING SIMULATORS.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 209-216
DOI: 10.5220/0001650102090216

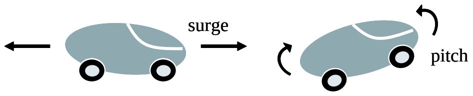

Figure 1: The longitudinal mode.

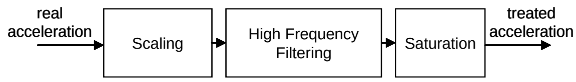

Figure 2: Preliminary treatments for the classical MCA.

of sensation to close the description of the simulator objective.

Remark: The results presented here are done for the longitudinal mode. However, they could be extended for the other space directions. The longitudinal mode is composed of the surge and pitch motions as depicted in figure 1.

2 MOTION PERCEPTION

Even in the absence of visual information (closed-eye subject), humans detect motion thanks to their inertial receptor: the *vestibular system*. Located in the inner ear, this biological apparatus measures both linear and angular motion of the head (a thorough description is given in (Elloumi, 2006; Telban et al., 2000; Angelaki and Dickman, 2004)) if they are beyond detection thresholds (on acceleration and speed respectively).

The *motion sensation* is built at the level of the brain not only from the vestibular system information but also from all the perception receptors (most particularly: the eyes) cues. In this paper, as commonly done in driving simulation, we consider that apart from the vestibular system, all the other sensors receive coherent and well adapted cues. As a consequence, in this paper the motion sensation will be considered as the interpretation of head displacements by the inertial receptor.

One remarkable gain of working with motion sensations instead of real trajectories (accelerations) is illustrated by the tilt coordination. In driving (or flight) simulation, a simultaneous rotation of the driver's head and the visual scene at a very slow rate happens to create an illusion of linear acceleration: "*When a visual scene representing an accelerated translation is presented to the driver while the simulation cockpit is tilted at an undetectable rate, the relative variation of the gravity vector will be partly interpreted as an actual linear acceleration*" (Reymond et al., 2002). Thus from a control point of view, the tilt coordination leads to a low-frequency motion sensation through a very small variation of the jacks' displacement as we shall see in the next section.

3 CLASSICAL MOTION CUEING ALGORITHM

This scheme was developed in 1970 by (Parrish et al., 1975). Despite its simplicity, this algorithm displays the importance of tilt coordination to restitute longitudinal accelerations. This scheme is based on the simple observation that the simulator translation is very limited so that only fast (*onset*) accelerations could be tracked. Consequently, the principle of this method is to use filtering to extract from the real car acceleration the high frequency component and address it to the robot translation. Hopefully, the tilt coordination enables the reproduction of slow (*sustained*) accelerations. Filtering (low frequencies) is performed to supply the tilt rotation as well. As for the restitution of the rotation speed, high pass filtering is performed to deal with angular limits.

The classical MCA is a linear approach which is commonly preceded by some preliminary treatments of the real accelerations to cope with robot motion limits (see figure 2).

4 THE REDUNDANCY PROBLEM

Restituting longitudinal acceleration on redundant simulators could be done thanks to three degrees of freedom (dof) as depicted in Fig.3: the base translation (X) (performed by the rails), the hexapod translation (x) and the tilt coordination (θ: tilt angle) (both performed by the jacks). As shown in (Elloumi, 2006), the behavior of the last dof is independent from the first two as the rotation due to the tilting is limited by a very low detection threshold.

As a consequence, in order to improve the quality of motion cueing only the translations behaviors should be considered. The considered linear acceleration[1] provided to the driver by the simulation robot is then: $\ddot{X} + \ddot{x}$.

Besides as the rails and jacks bandwidths are overlapping in the high frequencies domain, these two dof could be considered as equivalent. How could we

[1]The tilt coordination contribution $g\theta$ (where g is the gravity magnitude) is omitted from the hybrid algorithms that we shall present (but could be added outside these algorithms).

then exploit this equivalence? We offer two algorithms based on the hybrid systems framework which use only one translation at a time and a switching strategy to cope with the limits.

Figure 3 shows these two translations: hexapod translation (x) and base translation (X) each constrained by three levels of limitation: position, speed and acceleration $\pm\xi_L, \pm\dot{\xi}_L, \pm\ddot{\xi}_L$ ($\xi \in \{x, X\}$). The models ruling the variation of $\xi \in \{x, X\}$ are linearized models (double integrators):

$$\ddot{\xi} = u_\xi, \;\; \xi \in \{x, X\} \tag{1}$$

where u_ξ is the reference acceleration (control). As these dof are limited, we have to define two strategies: a braking strategy (triggered when nearing the limits) and a washout strategy (going back to a neutral position) strategy (once braking has been done).

4.1 Braking Strategy

In this paper we adopted the parabolic braking (constant braking acceleration $\ddot{\xi}_b$) in order to stop the translation at its limits $\pm\xi_L$ (null speed $\dot{\xi} = 0$). The triggering condition is then:

$$\dot{\xi}^2 - 2\ddot{\xi}_b(\xi_L - |\xi|) \geq 0, \;\; \xi \in \{x, X\} \tag{2}$$

The braking acceleration $\ddot{\xi}_b$ determines the free zone (braking-free zone) size. The higher $\ddot{\xi}_b$ is the bigger is the free zone. Nevertheless, the incoherent sensations would be strong in this case. At the opposite, choosing this acceleration to be as low as the detection threshold (0.05ms^{-2}) would considerably reduce the braking sensations and would noticeably reduce the free zone at the same time.

We have studied the influence of this parameter on the ratio between the free zone volume and the theoretically available one. As the phase profile (speed and position) is independent from the acceleration value, this ratio is equal to the ratio between the surfaces of the phase profiles. The theoretical surface is $S_{theo} = 4\xi_L\dot{\xi}_L$ and the free zone surface is:

$$S_{free} = \begin{cases} \frac{16}{3}\xi_L^{\frac{3}{2}}\sqrt{\ddot{\xi}_b} \text{ if } 0 \leq \ddot{\xi}_b < \frac{1}{4}\frac{\dot{\xi}_L^2}{\xi_L} \\ 4\left[\xi_L - \frac{1}{4}\dot{\xi}_L^2\ddot{\xi}_b^{-1}\right]\dot{\xi}_L + \frac{2}{3}\dot{\xi}_L^3\ddot{\xi}_b^{-1} \text{ otherwise} \end{cases} \tag{3}$$

The ratio S_{free}/S_{theo} is saturated starting from a certain braking acceleration $\ddot{\xi}_b$. In other words, starting from this point, the magnitude of $\ddot{\xi}_b$ wouldn't have a significant impact on the free zone size. However, the braking duration $\dot{\xi}_0\ddot{\xi}_b^{-1}$ (bounded by $\dot{\xi}_L\ddot{\xi}_b^{-1}$) will keep decreasing.

4.2 Washout Strategies

The goal of the washout is to bring the translation to its neutral position $(\xi, \dot{\xi}) = (0, 0)$. We present two washout strategies:

4.2.1 Known Starting Point

In this case the backward motion starts at $(\xi, \dot{\xi}) = (\pm\xi_L, 0)$, the chosen washout control is:

$$u_\xi = -\text{sign}(\xi)\begin{cases} a_r \text{ if } \frac{\xi_L}{2} \leq |\xi| \leq \xi_L \\ -a_r \text{ if } 0 \;\leq |\xi| \leq \frac{\xi_L}{2} \end{cases} \tag{4}$$

Taking $a_r = a_{threshold} = 0.05\text{ms}^{-2}$ would make this motion imperceptible. Finally, the duration of this strategy is: $2\sqrt{\xi_L a_r^{-1}}$.

4.2.2 Unknown Starting Point

In this case the backward motion starts at an unknown point (within the limits). The control is then a Proportional Derivative (PD):

$$\ddot{\xi} = -\mu\dot{\xi} - k\xi \tag{5}$$

The parameters (μ, k) have to be chosen so that the motion limits are respected.

These definitions of braking and washout techniques enable us to present our hybrid algorithms.

5 SYMMETRIC ALGORITHM

The principle of the symmetric algorithm is to use only one translation at a time. When the active translation reaches its limits, switching will be performed to activate the idle dof. In other words, both translations reproduce the reference acceleration as *relay runners*.

In non redundant simulators (without rails), when the hexapod translation is close to its limits, braking and washout will be successively triggered. The operator has to wait until these two operations finish in order to get back coherent motion cues. The symmetric algorithm will speed up the reactivation of the acceleration restitution by using the idle translation during the washout (impercebtible) motion of the active one. This algorithm is called *symmetric* because both translations have the same role in the motion cueing process.

In order to represent the symmetric algorithm as a hybrid automaton, two points have to be defined: the working modes and the rules of correct operation.

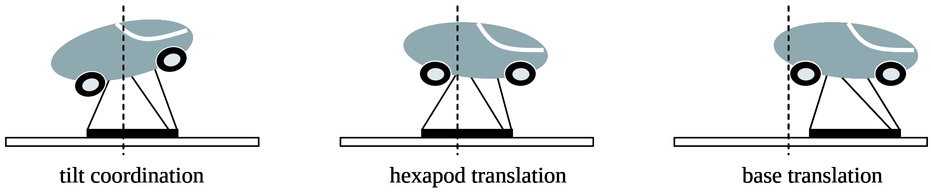

Figure 3: The redundancy of the longitudinal acceleration restitution.

Working Modes

In the case of the symmetric algorithm, both translations (X and x) have *the same* working modes:

1. *active*: the dof ξ tracks the *treated* reference acceleration
2. *brake*: parabolic braking
3. *washout*: known starting point
4. *idle*: null acceleration

Rules of Correct Operation

By defining these rules we characterize the way the hybrid automaton works. In our case, these rules are valid for *both* translations:

1. braking must lead to the limit position with a null speed
2. braking must be followed by a washout motion
3. washout must lead to the neutral position with a null speed
4. reactivating one dof could be done only starting from the neutral position (with a null speed)
5. if one dof is braking, the other one mustn't be active. The braking sensations could deteriorate indeed the quality of the free dof restitution

5.1 The Symmetric Automaton

Figure 4 shows the symmetric automaton (we observe a central symmetry around the state (**Hexapod:** washout, **Rail:** washout)). Three types of transition predicates appear:

- `hexa=0` (or `rail=0`) i.e. the dof has attained the neutral position $(\xi, \dot{\xi}) = (0, 0)$
- `decl_hexa` (or `decl_rail`) i.e. the braking condition is fulfilled (see (2))
- `lim_hexa` (or `lim_rail`) i.e. that an extreme position $(\xi, \dot{\xi}) = (\pm\xi_L, 0)$ has been attained

If we consider that braking is instantaneous then this automaton could be reduced to the four states outside the dashed box i.e. an alternation between the activation and the washout for both translations. The states inside the box take into account the parabolic braking and the subsequent activation of the washout.

5.2 Simulations

Matlab/Simulink and Stateflow were used to perform simulations. The initial state was chosen to be (**Rail:** active, **Hexapod:** idle). The reference acceleration profile was extracted from the Renault simulations in (Dagdelen, 2005). The parameters values used in these simulations are : $\ddot{X}_b = 1ms^{-2}$, $\ddot{x}_b = 0.2ms^{-2}$ and $a_r = 0.05ms^{-2}$ for both dof. Transition times are indicated by vertical lines. In figure 5, we can distinguish 6 working phases:

- (**Rail:** active, **Hexapod:** idle)
- (**Rail:** brake, **Hexapod:** idle)
- (**Rail:** washout, **Hexapod:** active)
- (**Rail:** washout, **Hexapod:** brake)
- (**Rail:** washout, **Hexapod:** washout)
- (**Rail:** active, **Hexapod:** washout)

6 MASTER-SLAVE ALGORITHM

In this algorithm the roles played by the two translations are *asymmetric*. One dof is the *master* i.e. responsible for restituting the motion. The other dof is the *slave* which has to counterbalance the "bad" master behaviors. It consists in producing opposite accelerations to the master's when the latter brakes or goes back to the neutral position (washout).

Working Modes

The working modes are different for each translation. The master's modes are:

1. *active*: the master dof tracks the treated reference acceleration

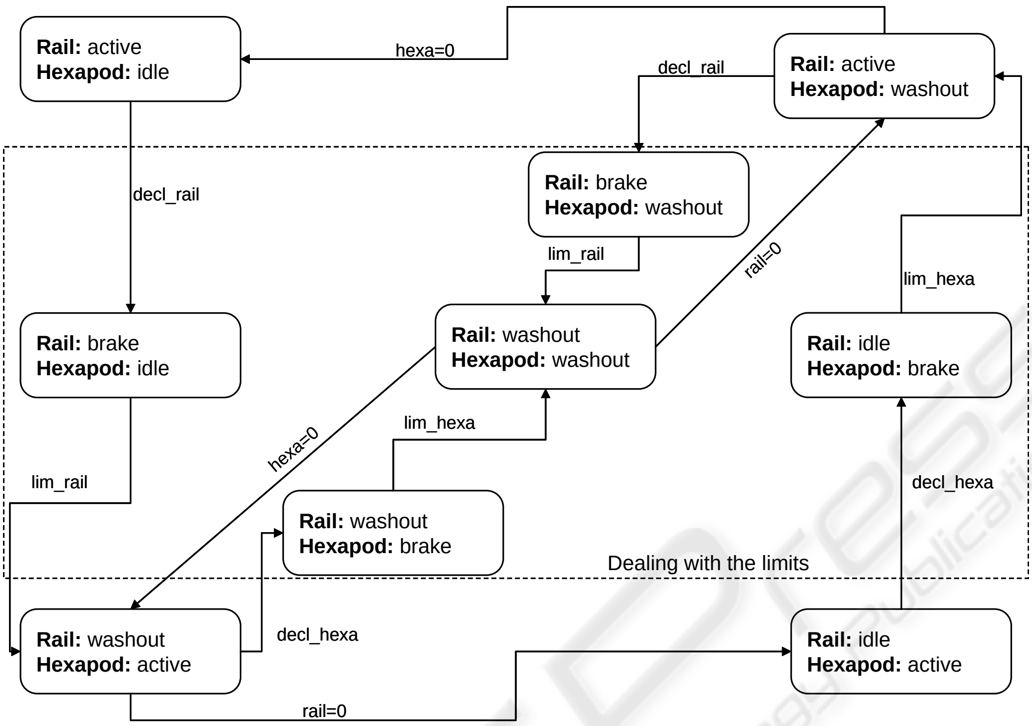

Figure 4: The symmetric automaton.

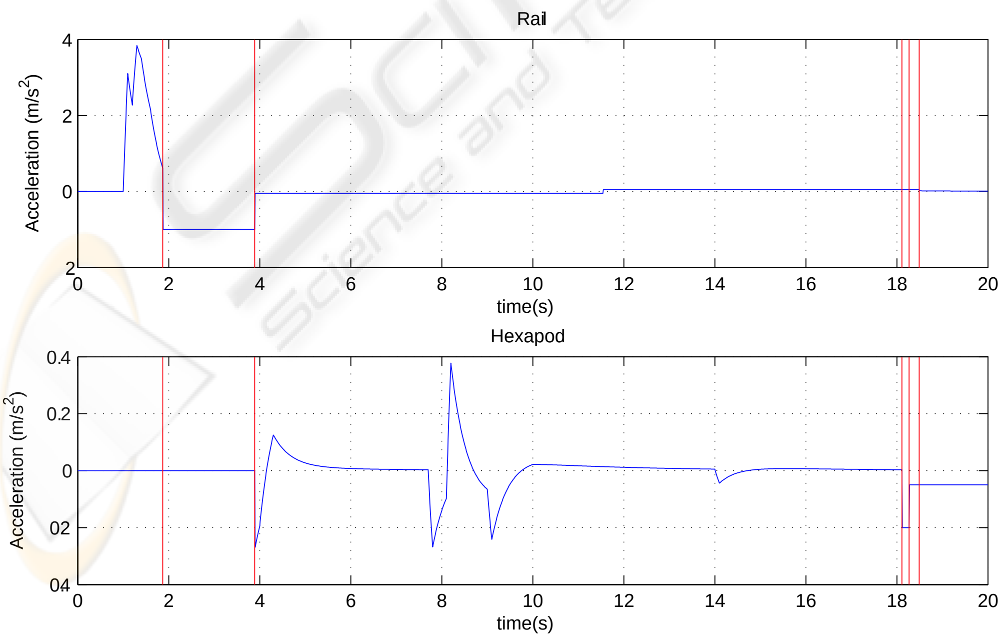

Figure 5: Simulation of the symmetric algorithm.

2. *brake*: parabolic braking
3. *quick-washout*: known starting point
4. *washout*: unknown starting point and undetectable
5. *idle*: null acceleration

The slave's modes are:

1. *counter-brake*: the slave dof tracks the acceleration opposite to the master's parabolic braking one
2. *counter-washout*: the slave dof tracks the acceleration opposite to the master's quick washout one
3. *brake*: parabolic braking
4. *washout*: unknown starting point and undetectable
5. *idle*: null acceleration

Rules of Correct Operation

1. braking of both translation (master and slave) must lead to the limit position with a null speed
2. braking must be followed by a washout motion. The master's washout is quick only if the slave is within its free zone
3. the master could be (re)activated only starting from its neutral position (the slave mustn't be in the braking mode)
4. after the counter-washout mode, the slave starts a washout motion to its neutral position

6.1 The Master-slave Automaton

Figure 6 shows the master-slave automaton. The transitions have the same signification as for the symmetric automaton. Similarly, if we consider that braking is instantaneous then this automaton could be reduced to these subsequent states:

- Initial state: (**Master:** active, **Slave:** idle)
- (**Master:** brake, **Slave:** counter-brake): when nearing the limits, the master brakes. The slave provides the opposite acceleration so that the total acceleration (perceived by the driver) is null.
- (**Master:** quick-washout, **Slave:** counter-washout): after the master's braking, a quick motion brings it to its neutral position. The slave counterbalances this motion so that the overall acceleration is null again.
- (**Master:** active, **Slave:** washout): the master is reactivated once reaching its neutral position. The slave starts the washout in order to improve its future capacity of compensation. From this state two permutations could occur: going back to the initial state if the slave washout has finished or switching to (**Master:** brake, **Slave:** counter-brake) if the master reaches once again its limits.

The dashed box integrates all the states that describe the automaton behavior when the slave couldn't perform its counterbalancing role. It happens when the slave reaches its limits and has to break. In this case, the hybrid automaton starts a backward motion of both dof that ends by going back to the initial state.

6.2 Simulations

The reference acceleration profile is the same as before (scaled at 50% for a better visualisation). Rails were chosen to be the slave whereas the hexapod translation is chosen to be the master. In fact, as the rails motion capacity is higher than the hexapod one, the former is better suited to play the compensation (slave) role.

The algorithm parameters are: braking (and counter-braking) acceleration $0.3ms^{-2}$, quick washout (and counter-washout) acceleration $0.5ms^{-2}$ and slave braking acceleration $0.6ms^{-2}$. As for the slave washout, μ and k were chosen to be τ^{-1} and τ^{-2} where $\tau = 1.45s$. Figure 7 shows the simulation results. We can distinguish 5 phases:

- (**Hexapod:** active, **Rail:** idle)
- (**Hexapod:** brake, **Rail:** counter-brake)
- (**Hexapod:** quick washout, **Rail:** counter-washout)
- (**Hexapod:** active, **Rail:** washout)
- (**Hexapod:** active, **Rail:** idle)

7 CONCLUSION

In this paper we have presented two motion cueing algorithms based on the hybrid systems framework. These two algorithms exploit the redundancy of the simulators to maintain the reproduction of motion sensations despite the robot displacement limitations.

The symmetric algorithm presents a reliable initial restitution. However it generates incoherent sensations due to significant braking magnitudes. The Master/Slave algorithm has a lesser restitution capacity but it reduces considerably bad sensations by providing a full compensation (null sensations at the level of the driver) of braking and washout motions.

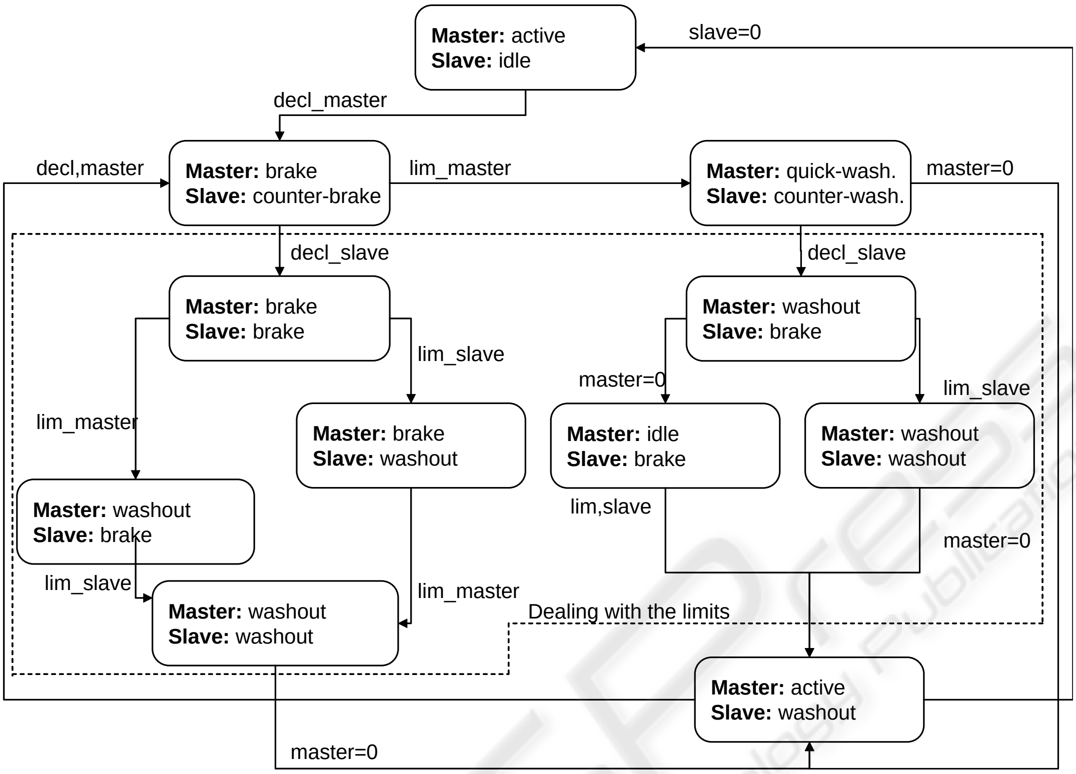

Figure 6: The master-slave automaton.

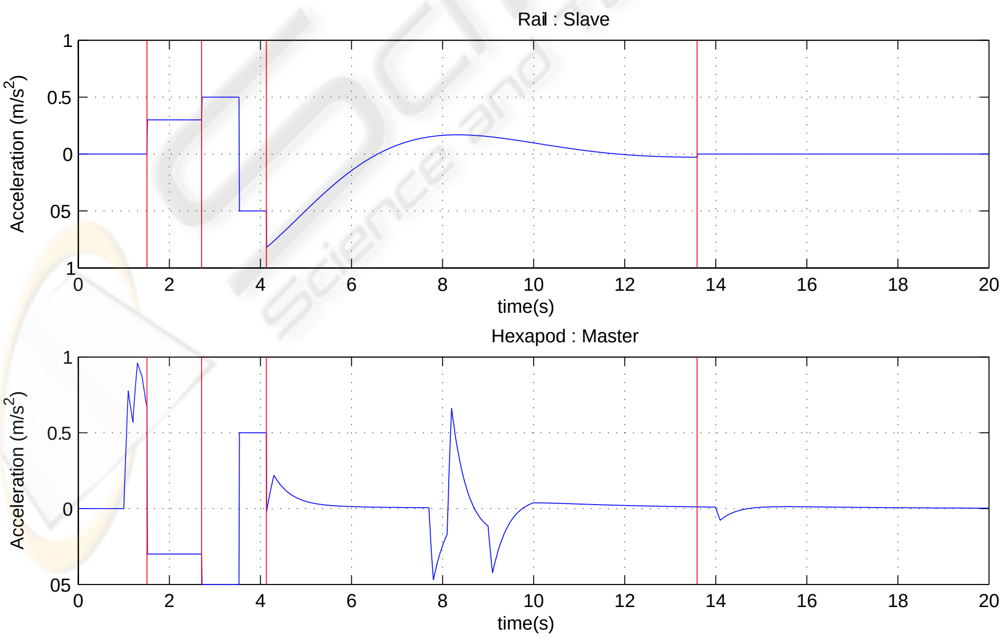

Figure 7: The master-slave simulations.

REFERENCES

Angelaki, D. and Dickman, J. (2004). Gravity or translation: Central processing of vestibular signals to detect motion. *Journal of Vestibular Research*.

Dagdelen, M. (2005). *Restitution des stimuli inertiels en simulation de conduite*. PhD thesis, Ecole des Mines de Paris - Renault.

Elloumi, H. (2006). *Commande des plate-formes avances de simulation de conduite*. PhD thesis, Ecole des Mines de Paris, Centre de Mathématiques Appliquées.

Merlet, J. (2000). *Parallel Robots*. Kluwer Academic Publishers.

Parrish, R., Dieudonne, J., and Martin Jr., D. (1975). Coordinated adaptive washout for motion simulators. *Journal of Aircraft*.

Reymond, G., Heidet, A., Canry, M., and Kemeny, A. (2002). Validation of Renault's dynamic simulator for adaptive cruise control experiments. *Driving Simulation Conference*.

Telban, R., Cardullo, F., and Guo, L. (2000). Investigation of mathematical models of otolith organs for human centered motion cueing algorithms. In *AIAA Modeling and Simulation Technologies Conference,* Denver CO.

Van der Shaft, A. and Schumacher, H. (1999). *An introduction to hybrid dynamical systems*. Springer-Verlag, Lecture Notes in Control, 251.

Zaytoon, J. (2001). *Systèmes dynamiques hybrides*. Hermes Sciences Publications.