CHAPTER 4

Optimal Mechatronics for Driving Simulator Design

Lamri Nehaoua, Hichem Arioui, Nicolas Séguy,
Informatique, Biologie Intégrative et Systèmes Complexes
Evry University
Evry, 91020, France
{nehaoua,arioui,seguy}@ibisc.univ-evry.fr

ABSTRACT. This chapter discusses the necessary conditions for the successful design of a driving simulator. This success is assessed, among other factors, by the quality of the rendered motions, when driving the simulator.

We discuss throughout this chapter, the philosophy of driving simulation to better explain the challenges of these applications. Particular attention will be paid to the interaction between the various entities, making up a simulation system operational. This interaction is intended to highlight the importance of mechatronics in the successful design of such a simulator.

Keywords: Driving Simulator, Mechatronics, Optimal Design.

4.1. Introduction

The concept of driving simulators existed as far back as the early 1970s. At that time, considerable research on safety was performed focusing on vehicle dynamics, in which stability issues were important. A dynamically-correct simulator should be something that could be used to study how different vehicle parameters influence stability.

Simulators were mainly used in aviation and then primarily for training in the use of cockpit instruments. In a driving simulator, the driver does not rely on instruments to the same extent. Here, the surroundings and dynamic forces are more important. It soon became clear that a high fidelity driving simulator requires a sophisticated motion simulation, a detailed model of the vehicle's dynamics and a visual description of the road environment.

The achievement of a driving simulator is a serious multidisciplinary challenge, because each driving simulator is a single prototype and there is no specific standard design. The success of such implementation can be reached with the cooperation of different skills and experts, from the designer to the final user. Therefore, the realization of such a platform requires a design at various levels of abstraction in order to address the best different specifications and constraints. It is obvious that the achievement of each level will involve limited choices in finding the good balance between the various hardware and software components.

A driving simulator is considered as a complex mechatronics system. It is composed, among other components, of a mechanical platform and an embedded electronics for its functioning. Its design should follow the approaches of modeling and simulation of multi-physics to extract an optimal solution. Previously, the choice of the optimal architecture was done by seeking the most appropriate solution for each component. The performances of the whole are checked by a juxtaposition of the different blocks. Otherwise, and following a mechatronics approach, the optimal design of a driving simulator requires a multi-physic modeling and optimization of the system in its entirety. However, in practice, such an approach is still not feasible. Thus, the problem is divided into sub-problems that can be optimized separately. Then, each solution is evaluated and adjusted to provide a sub-optimal architecture.

The quality of a driving simulator is assessed by its fidelity level, from a perceptive point of view; to better feedback the car motions into the driver. This restitution is largely dependent on the platform of mechatronics (mechanics, embedded electronics, actuation's technologies and control laws). Thus, an optimized mechatronics strongly helps the system to provide optimum motions, conversely, a non-synchronized mechatronics induce perceptive errors to users or can even make them feel very uncomfortable.

FIGURE 4.2.1. University of Padua Riding Simulator

4.2. Overview on existing simulators

The literature reveals that numerous driving simulators exist in the world. Whether academic, industrial or commercial, many institutions have built their own vehicle prototypes for different purposes [**1, 2, 3**]. However, the bibliography on two-wheeled riding simulators is sparse. The main works were done by Japanese and Italian industrial institutions. In 1988, HONDA Corporation started to develop series of motorcycle simulators. The first prototype of a dynamic platform was designed to test the feasibility of driving simulation to reproduce the basic maneuvers of a motorcycle dynamics. The mobile platform has 7 actuated axes to simulate 4 Degrees of Freedom (DoF) including roll, yaw, pitch, and handlebar steer. A cradle mechanism was developed to simulate the feeling of sustained acceleration. Next, a second prototype was developed and installed in the center of traffic education at Suzuka since 1991 to assert the simulator's effectiveness as an approved training tool [**4, 5**]. The architecture of the new platform was completely modified. The cradle system was removed and only 3 DoF were retained: pitch ($\pm 10°$), roll ($\pm 15°$) and steer ($\pm 30°$).

Outside Japan, a simulator was born from the collaboration between the PERCRO laboratory and the motorcycle manufacturer Piaggio. Appointed as a rapid prototyping tool, it is based on a 6 DoF mechanical parallel platform, hydraulically actuated. A real scooter chassis is mounted on the mobile platform [**6**]. In the same way, a bicycle simulator was built at the Korean Advanced Institute of Sciences and Technologies (KAIST). The motion generation is also ensured by a 6 DoF mechanical platform electrically actuated on which a bicycle frame is fixed. The handlebar and the pedal are respectively equipped with active and passive haptic devices [**7**]. Based on a serial mechanical architecture, a riding simulator is designed at the department of mechanical engineering at Padua University [**8**], Figure 4.2.1. This simulator allows the simulation of 5 DoF including roll, pitch, yaw, lateral displacement and steer angle.

Within the framework of the French project SIMACOM, a riding motorcycle simulator was conceived in collaboration between the National Institute of Research in Transportation and Safety (INRETS) and the laboratory of Informatics, Integrative Biology and Complex Systems (IBISC) France [**9**]. It allows the simulation of 5 DoF, namely, roll, pitch, yaw, steer and arms displacement, Figure 4.2.2.

4.3. Single/Double track dynamics

For a driving simulator, the dynamic model of the vehicle is responsible for generating reference trajectories to move the mechanical platform. Accordingly, the platform computes the new states of the virtual vehicle in response to the various actions of the driver.

In theory, a vehicle has six degrees of freedom (DoF) consisting of three translations and three rotations, Figure 4.3.1. The translation on the X axis denotes the longitudinal displacement, along the Y axis is the lateral displacement and vertical translation is done along the Z axis which reflects the movement of the chassis. The rotation around the Z axis is the yaw ψ of the vehicle that determines its trajectory, a second rotation called roll φ around the X axis defines the inclination of the body when taking a turn or lane change. Finally, the rotation ψ around the Y axis describes the pitching of the vehicle encountered during acceleration and braking phases.

Over its movement, most of the external forces acting on a vehicle are generated at tire-ground interface, Figure 4.3.2. Initially, we can identify the lateral guidance force, longitudinal force of traction / braking, the vertical one and

FIGURE 4.2.2. IBISC-INRETS Riding Simulator

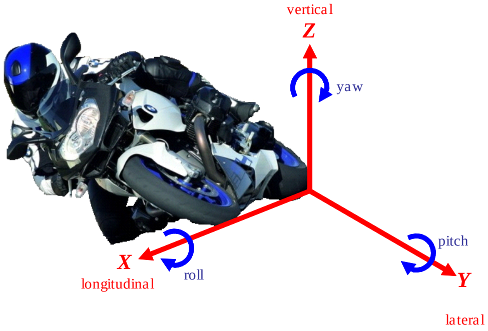

FIGURE 4.3.1. Single track vehicle motions and frames

their corresponding moments [**10**]. Although these forces are strongly linked, they are generally processed separately. On the other hand, aerodynamic actions consist of the longitudinal force, lateral thrust and vertical lift applied at the center of gravity. Beyond these efforts, of external origin, others are of intrinsic nature and result primarily from various connections and mechanical knowledge of the vehicle such as: suspension, steering, ... etc.

The complexity of the dynamic model depends on: (1) the mechanical architecture of the driving simulator and (2) the nature of the application to carry out. For example, if the simulator is used for automotive applications, the dynamic model must be dynamically rich. Otherwise, representations able to transcribe the lateral and the longitudinal dynamics are sufficient.

Considering the motorcycle as a set of rigid bodies connected by simple joints, multi-body system mechanics theory offers a convenient framework to derive its motion model. In [**11**], authors have adopted Lagrange formalism to derive their motorcycle dynamic model. A direct application of this approach, leads to unattractive performances in terms of the number of operations and implementation facilities. In Khalil [**12**] and Hollerbach [**13**], Recursive Newton-Euler Algorithm (RENA) was shown to be effective, Figure 4.3.3. Originally, this technique was developed to solve the inverse dynamic model of open chain manipulators with a fixed base for control purposes. By projecting the dynamics of each body in its attached frame, the acceleration of the joint variable can be easily derived without the need to inverse the whole system inertia matrix. In addition, this technique, named Articulated Body Algorithm (ABA), is more numerically stable than the inertia matrix inversion method.

As an alternative to this approach, a more flexible algorithm has been presented in [**14**]. In this model, the motorcycle is considered as the saddle body, the front upper body (handlebar and upper part of suspension), the front lower part (lower part of the suspension), the swinging arm and the two tires. The handlebar and the swinging arm are attached to the saddle body by a simple revolute joint. The front lower part is linked to the front upper part by a prismatic joint. In addition, the rear suspension is connected to the saddle with a revolute joint from one side and to

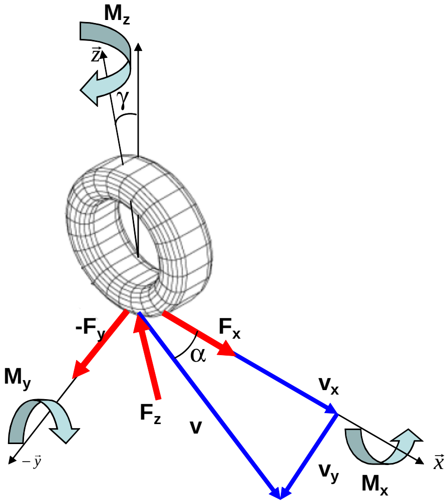

FIGURE 4.3.2. Efforts at tire-road interface

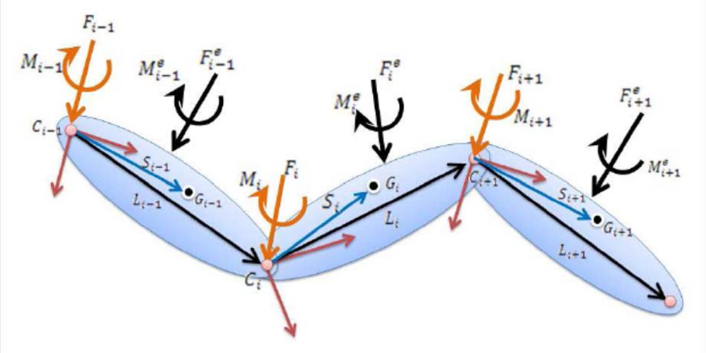

FIGURE 4.3.3. Open chain multi body system

the swinging arm with a revolute joint also on the other side. This creates a closed kinematic loop leading to solving difficulties. Finally, the rear and front wheels are respectively connected to the other tips of the swinging arm and front lower part, as sketched in Figure 4.3.4.

Furthermore, an engine model is often very useful, even indispensable. Its goal is to compute the engine power transmitted to the vehicle based on the driver actions on the pedals: throttle, clutch, brakes and gearbox selector. To establish a model close to a real behavior, one should take into account the thermodynamics equations related to flows and combustion phenomena. Furthermore, although the engine seems to have a continuous functioning, it is actually a hybrid system with a succession of cycles almost independent (compression, expansion, etc.). Some engine models already exist in the literature, but they can be implemented on powerful computers. Therefore, they are not suitable for driving simulation applications. Nevertheless, some simple models can be adjusted for real-time operations.

4.4. Design and mechanical aspects

During the design phase, a first consideration should be conducted for all sub-systems with a central question: "what do we need to reproduce to the driver?" according to the planned tests and the application for which the simulator is designed. For example, for the straight line driving case, a vibrating table is largely sufficient. For other situations, a more sophisticated mobile platform may be required.

First, simulator design requires the definition of an appropriate mechanical architecture. An intuitive choice is to opt for a parallel platform like Gough-Stewart's. The advantage of these platforms is that they cover the 6 DoF

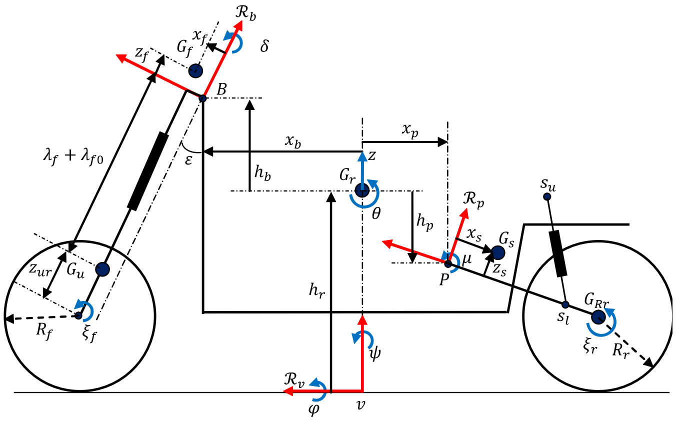

FIGURE 4.3.4. Motorcycle kinematic configuration

with the possibility to choose the instant center of rotation, which position relative to the system of driver perception (vestibular system), seems important. However, the price of this solution is far from being affordable to be supported by the users.

The choice of driving simulator architecture is guided by the need to ensure a sufficient perception level, previously defined. The purpose of driving simulator applications relates generally two distinct frameworks that are: (1) risk training to make drivers aware of dangerous situations and (2) drivers' behavioral observation in normal driving, especially in urban situations. Simulator's users, can as well be training centers or road safety agencies. The system must, therefore, be of an acceptable cost and easily transportable. Thus, it is necessary for every application to identify driving situations which should be considered. Risk training aims at increasing the awareness of new drivers to road risks. By motorcycle, some situations are unassailable, as the skidding of the front wheel; the simulator must still allow the driver to adjust himself to this situation. In this case, the transit time stability to instability is very short; therefore, the reproduction of this behavior involves the performance of an actuation system. While the purpose of behavior observation is to enable researchers to understand the cues perceived by the driver, in order to develop assistance system or to test the interaction with a special infrastructure. Extreme cases should therefore be produced as well as cases of normal driving.

For both previous scenarios, the learning context should be preferred to the faithful reproduction of the entire motions or risks. The proposed platform architecture (workspace, dimensioning of actuators, etc.) should promote this approach. This is consistent with the constraints of developing a low-cost tool. Under these considerations, the degrees of freedom are fixed. For a single track vehicle, three motions should be reproduced. The roll motion is considered to be the most important one. This degree of freedom is essential for stabilizing and leading the motorcycle. It is also involved in cornering manoeuvres, slalom and lane changing. Then, the pitch motion is reproduced to feedback rear longitudinal acceleration, braking phases and fork dynamics. Finally, the yaw motion may be specifically used in order to reproduce skidding phenomena. The skidding of the front wheel is not interesting because it has fatal consequences.

In addition, we know that the multiplication of perceptual stimuli can strongly increase riding simulation sensations. Based on this idea, one can add passive/active device so that the driver is well "‘immerse"’ in his virtual environment. For example, a force feedback system can be implemented on the handlebars, Figure 4.4.1. The aim is to create other transitory phenomena to feed back to the user: inertial cause on bust during acceleration and braking. Thus, an effort is created in the arms of the rider by varying the distance between the saddle and the handlebar. Another force feedback can be developed to render the resulting torque of the tire-ground on the steering axis of the motorcycle.

At the kinematic level, the position of the different axes of possible rotations is crucial. At the authors' knowledge, no psychophysical study has been conducted, except in some special cases [**5**].Therefore, these axes are defined from the kinematics of a real motorcycle. In [**9**], the authors reproduce the necessary yaw in order to feel the rear wheel skidding, a slide system is placed on the back of the motorcycle frame. The roll axis is placed in the motorcycle symmetry plane with an adjustable height in order to test various configurations and to achieve the best perception

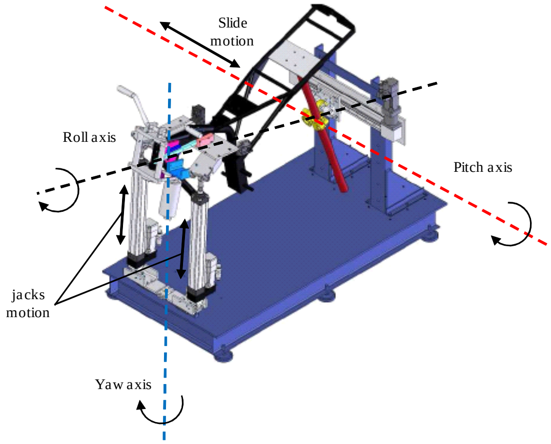

FIGURE 4.4.1. CAD model of the mechanical platform with its different rotation axes

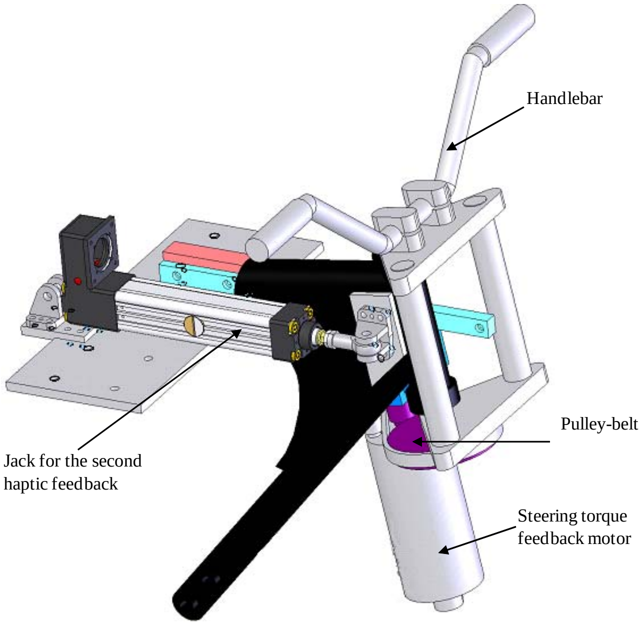

FIGURE 4.4.2. Haptic device for torque feedback on handlebar

results. Lastly, for the pitch axis, it is the displacement of the front fork in the acceleration and braking phases which were privileged, therefore this axis goes through the back of the motorcycle frame, Figure 4.4.2.

4.5. Platform instrumentation

Actuating the simulator's mechanical platform amounts to setting its states of motion in reply to driver's actions. This results in the generation of so-called reference trajectories that are in adequacy with the user's actions, actuators performance and the platform mechanics. All inputs to this mechatronics system should be sent from either form of sensory transducers and sub-blocs communication. To that effect, the simulator is equipped with conventional motorcycle controls, with associated sensors. Information from these sensors is also considered as inputs to the virtual motorcycle and summarized as following:

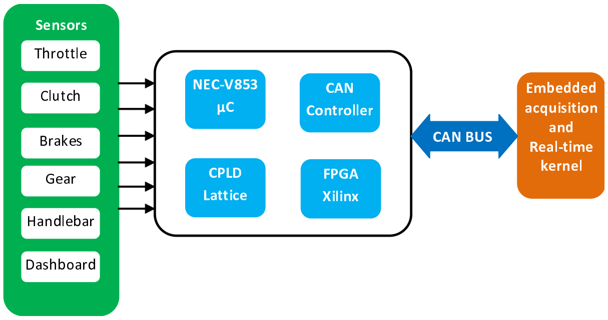

FIGURE 4.5.1. Rider's action instrumentation. Sensors, acquisition board and communication

- The position of the throttle and clutch lever: these inputs serve for computing the engine output power. Throttle and clutch actions are instrumented by using simple linear resistive sensors known as potentiometers. This last consists of three terminals where the second one is connected to a sliding wiper attached to each level. Analog electrical information, image of the lever position, is delivered.
- The braking power of the front brake lever is returned through a pressure sensor that is mounted on the hydraulic assembly. This sensor delivers an analog electrical signal, image of the rider's torque exerted on the lever's handle. The braking power of the rear brake pedal is provided with the same device as well.
- Gearbox information: the gearbox signal translates the shift state used to update the engine speed. For this, a mechanical switch (SPDT: single pole double throw) is mounted beside the speed selector which returns binary information. A typical position switch with a mounted lever creates a mechanical contact when a shift is engaged which closes the electrical circuit within the switch. A logic circuit based on bascules, is included to prevent the signal from bouncing resulting in lever micro-impacts.
- The handlebar position: theoretically, two-wheeled vehicles are controlled mainly by the steering torque applied by the rider on the handlebar.

However, the cost generated by the implementation of a torque sensor is in contradiction with a low-cost design philosophy. Recently, control theory has provided engineers with powerful techniques allowing implementation of virtual sensors, namely, unknown inputs observers. These observers can perform, starting from some measurements, the estimation of an unknown information signal like the rider torque exerted on the motorcycle handlebar. To achieve this, the handlebar's position is an important measure which should be acquired by a maximum of precision. To that end, a common sensor for position measurement is the optical incremental encoder providing digital information with a given precision, 1024 points in the current simulator. Nevertheless, whatever the simulated manoeuvres, the steering angle is infinitely small and therefore one must choose a high resolution sensor. Another simple solution is to trade on the available sensor dynamics by using a pulley-belt system. This solution is more practical and advantageous to also implement a torque feedback on the handlebar.

For the present motorcycle simulator, the acquisition of the rider actions is ensured by a set of two plugged electronic cards forming a compact acquisition board. The first provides an I/O interface for sensor signal routing, while the second deals with the signal low-level management. This board consists of a micro-controller V853 from NEC Corporation with 12 analog inputs (± 10Volt), 16 analog outputs (0-5Volt) as well as multiple digital I/O. Binary buffered inputs dedicated to the acquisition of the optical encoder channels are managed by an FPGA (Xilinx XCS30XL). Intercommunication among host computers is proposed in various flexible configurations ranging from the parallel port to CAN (Controller Area Network) communication frame.

The overall architecture of an embedded system (sensors, acquisition, data processing, decision making, and actuation) requires special attention with respect to the data exchange flow and requirements on the data reliability level. Signal alteration, sampling and protection against noises must be taken into consideration. According to the number of users and mechatronic sub-systems, point-to-point or point-to-multipoint communications are possible. A distributed vision of such architecture constitutes an initial guess to reduce its complexity. Indeed, a multiplexed bus, type CAN, reduces the level of coupling between different components. Thus, for the present simulator, the acquisition is accomplished by establishing communication via the CAN bus by an integrated controller.

4.6. Actuator selection and driving

The choice of actuators is in line with the expected performance in terms of perception and ability to put the rider into a risk situation. This is done by full study of the stability of a two-wheeled vehicle. Indeed, the analysis of motorcycle dynamics gives rise to three significant, well separated, modes of instability [**15**]. The "Capsize" is related to the roll motion where gyroscopic effects turn out to be negligible to stabilize the vehicle. It is a non oscillatory, well damped mode at low speed, but, beyond a speed of 20 (m/s), this mode becomes unstable and can be controlled by a steering torque applied on the motorcycle handlebar. The "wobble" is a fast oscillating mode involving the steering motion of the front fork, whose frequency range into 4-10 (Hz). Lastly, the "Weave", is a side-to-side motion of the entire motorcycle involving yaw and roll oscillations with significant lateral displacement. Weave is an unstable mode at very low speed (less than 5 (m/ s)) and well stable at high speeds over 30 (m/s). Its frequency ranges between 0 and 4 (Hz), and it affects the entire two-wheeled vehicle which makes it difficult to be controlled by the motorcycle rider. These modes are at the limit of the vehicle stability and beyond, the rider lands with a falling and risk situations. Moreover, the two-wheeled vehicle is more powerful than a car vehicle with an important power/mass ratio leading to considerable accelerations.

To achieve the expected performance as prescribed in the simulator specifications, it is necessary to choose the platform mechanics (number of DoF) and the corresponding drive system. Advances in industrial computing and power electronics together have promoted the use of electrical machines over hydraulic ones. Indeed, hydraulic actuators present higher strength characteristic where the density of energy is almost 100 times greater than that of an ideal electric machine. But, this factor should not alone dictate the choice of the platform actuation, the versatility use, installation speed and high speeds of the electrical power with a better value for money, make the electrical actuators a convenient choice. Finally, the control problem is much simpler to lay with electrical drives. Hydraulic actuators exhibit strong nonlinearities marked by hysteresis. This fact is important and impacts the performance, precision and deployment.

If the possibility of a hydraulic actuator is ruled out, the platform actuation is to choose from a wide range of electric motors [**16**]. They can be classified based on functionality, rotor design and induced electromotive force (emf). The direct current (DC) brush motor consists of a stator and a rotor with coil windings (armature). The stator, formed by electro-magnets, creates a fixed magnetic field, thereby generating a very regular torque. The rotor winding is driven by a DC current which can cause the rotor rotation under the action of the well known Lorentz's electromagnetic force. Its simple nature gives him the advantage of being easy to control.

Synchronous brush motor looks like the previously described DC brush motor. Its rotor, powered by DC current through its brushes, creates a magnetic field that is in phase with the stator field. So, the rotor rotates at the same speed as the rotating field, hence its name "‘Synchronous motor"’. Furthermore, the permanent magnet synchronous motor (PMSM) is a synchronous motor where the conventional electromagnetic field poles in the rotor are replaced by permanent magnet poles yielding to a brushless motor. In this type of actuator, slip ring and brush assembly are dispensed with the help of an electronic switch enabling self-drive.

PMSM can be broadly classified according to the different ways the magnets can be arranged on the rotor. Magnet disposition has a direct impact on the flux density, winding inductance, reluctance torque and the shape of the induced emf, i.e., sinusoidal and trapezoidal, so:

- If the rotor magnet poles (also called smooth poles) are disposed, the induced emf is sinusoidal. In the literature, the sinusoidal type is shortly known as PMSM,
- If the rotor is of concentrated coil-wound (also called salient poles), the induced emf is trapezoidal. The trapezoidal type is called PM DC brushless motor,

The PM DC brushless motor with induced trapezoidal emf is driven by the six-step switching method where, the rotor phases are excited by 120° wide currents. The drawback of this method is that it produces large torque ripple. However, its main advantage is that the commutation signals need to be generated only six times for every electrical cycle. So, implementing a position sensor like Hall effect is very trivial. In addition, for a motor with sinusoidal emf, the controller aims to drive independently each phase current depending on the rotor position. This electronic switching method allows minimizing the produced torque ripple and improving the efficiency and power factor. However, controlling this type of motor is more difficult. Indeed, the drive system of the PMSM with sinusoidal emf needs the position information continuously in order to construct winding currents. Consequently, the precision of the current reference signal depends mainly on the position sensor resolution (optical encoder or resolver) which makes the cost of a PMSM higher against a PM DC brushless motor.

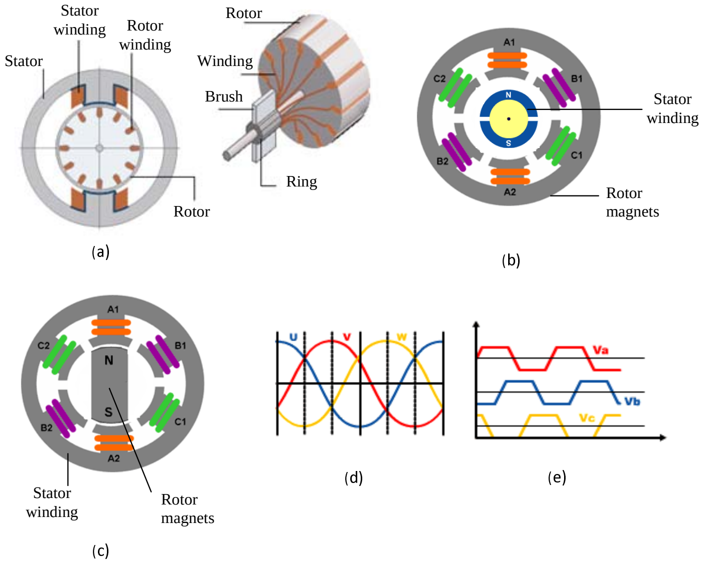

FIGURE 4.6.1. Electric machine classification : (a) direct current (DC) brush motor, (b) permanent magnet synchronous motor (PMSM) with one pair of smooth poles, (c) PM DC brushless motor with a pair of salient poles, (d) trapezoidal induced electromotive force, (e) sinusoidal induced electromotive force

Finally, the asynchronous motor is the most widespread in industrial applications like ventilating and pumping, where there is a need for constant speed driving. Following the restrictions imposed by the simulator specifications and in a spirit of a low cost development, the choice of the platform actuation seems to be obvious. To move the different axis of motion of the simulator's mechanics, a PM DC brushless motor with a trapezoidal emf is adopted. On the one hand, this actuator is well adapted for low positioning performance, has about 15% more density than the PMSM with a lower cost. On the other hand, for the handlebar steering torque feedback, the ease of integration and the absence of torque ripple are the two mandatory criteria to be taken into account and hence, a DC brush motor is selected.

The use of brushless motors is the widespread norm, thanks to the development of power electronic converters. In an electromechanical system, variable frequency of the stator winding current is needed to achieve variable speed. For this, the designer uses inverters to convert the reference signals as delivered by the controller to an electric current which produces the actuation torque. So, it is essential to develop the associated drive electronics. This drive includes several subsystems ranging from alternative current (AC) supply rectifier to low-level current servoing. Thus, a major question arises; do we use a commercial drive or a custom home solution?

It is difficult to answer this question and a compromise must be reached. A commercial drive provides a fast and efficient way for prototyping and control. Nowadays, in addition to the inverter and the switching circuit, drives are allocated with embedded devices allowing position, speed and torque control. However, a drive is not a flexible solution for the implementation of advanced control laws. As an example, for torque control, one should make sure that the drive allows the measurement of phase currents and the absolute position of the rotor. Nevertheless, if a conventional proportional, integral and, derivative (PID) control scheme should be implemented, a commercial drive is an appropriate choice.

A custom home solution provides a flexible, powerful and an open control device. However, torque control requires high acquisition speeds (over 500 (Hz)) and the current control loop should be performed at a frequency higher than 8 (KHz) (depending on the power components that constitute the inverter). Moreover, in the industrial domain, the use of three-phase AC current supply is predominant. It follows that a custom solution requires designing a rectifier, inverter, switching circuit, current sensor, signal conditioning, protection, insulation and implementation of the control algorithms as close as possible to the switching devices (the use of a DSP is mandatory). This solution, being more flexible, requires a greater investment and resources.

4.7. Real-time monitoring, sequencing and synchronization

The simulator is a set of mechanical, electronic and software components that must communicate with each other with respect to prescribed temporal constraints. All tasks must be performed in a binding manner, real-time and delay free. Therefore, synchronization is a key element for the simulator's operation. The problem of delays may lead to a loss of controllability of the simulator and a poor motion rendering.

The need for real-time is essential and refers to two main aspects which are event management and control of the simulator. These two aspects should be performed in a predefined time entity and hence, rethinking and reconceptualizing a dedicated platform is inevitable. Indeed, in the initial conceptual design, most of the development is carried out using a simulation tool like Simulink under Microsoft Windows operating system or a similar platform. Such an approach is well suited for rapid prototyping and requires much less time. However, Simulink uses block diagram representations and a sequential time vector to evaluate the different model states and outputs. Since this vector is not connected to the system clock, the outputs are determined in a non real-time way following the performance of the used computer. So, a real-time implementation requires that the programmer has control of all events; semaphore resources and can assign a priority to each process defined as the most critical. This commitment cannot be managed without an appropriate operating system which facilitates access to CPU resources, management processes, memory, interrupts and so on.

Recently, many professional and free license operating systems with a real-time kernel have been proposed. However, the major difficulty lies in the development of drivers used to interface the hardware components to the real-time kernel. Therefore, the use of a professional real-time manager is recommended for rapid development. In this perspective, according to the suitable final deployment, one can cite for example:

- xPC Target: a toolbox from Mathworks intended for MATLAB/Simulink developers. The advantage of xPC Target is its user-friendly and its trivial configuration. File generation, compilation and loading are done in an implicit manner with a minimum of intervention from the user. An equivalent version for Windows, called Real-Time Windows Target, also exists [**17**].
- LabVIEW Real-Time Module Development: A real-time package developed by National Instruments.
- Dspace: widely used in the automotive field, it consists of an acquisition host computer and a real-time kernel manager. All models to be simulated must be converted into C (C++) language. The main advantage is that dSpace is compatible with the Matlab/Simulink. All Simulink models can be converted into C language using an appropriate package like Target-Link or Real-time workshop toolbox. The major drawback of this tool is its price.
- Linux Kernel: Thanks to its stability and reliability, Linux is becoming more and more used in industry and application domains. Several real-time kernels have emerged as result of continuous efforts of a growing open source community. Thus, numerous solutions have emerged in the free license world such as Open RTLinux and RTCoreBSD. Professional versions, such as VxWorks, offer an adequate working interface and a wider portability.

Furthermore, the control task is a central element in the actuation of the platform mechanics. It consists of the trajectory generation, loop control and supervision. So, these three control steps will convert the states of the virtual vehicle, images of the rider's actions, in an actuator signals which drive the various motion axes. Otherwise, supervision is a task that includes, among system initialization, the handling of different events, the fault detection and diagnosis. In this sense, each motion axis (jacks and slide) is instrumented with two Hall effect switch sensors. These sensors provide binary output indicating that a joint has reached the limit of its allowable travel. In this way, it is possible to scrutinize the possible excursion and to avoid any unforeseen damage and preserving the platform mechanics. In addition, supervision task allows a stable connection between the visual rendering computer and the real-time kernel computer. Therefore, planning semi-static data exchange in the embedded network can provide a common level of dependability. In this context, the use of a multiplexed field bus is entirely justified.

According to this perspective, a monitoring interface based on a state machine is developed. This machine contains the essential processes to be implemented, as summarized in the following points:

- At the beginning of the simulation, the actuators are set to state 1 "Login". A CAN request, with the given identifier, is sent to each drive to set it into a control mode.
- The second step involves positioning the simulator's mobile platform to the neutral position (jacks and slide at mid-travel). This position will be considered as a reference to any future movement of the platform. At this

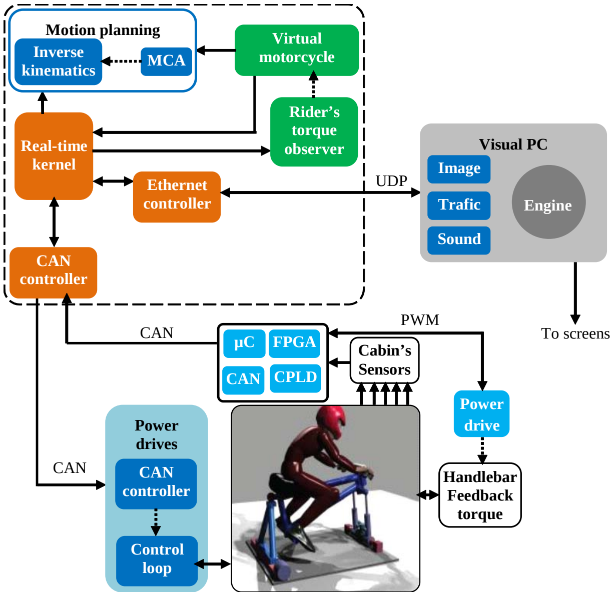

FIGURE 4.7.1. Overall motorcycle simulator mechatronic architecture

stage, essential information such as: position switch information and the maximum allowable workspace, will be available. Positioning is achieved simply by scrutinizing the displacement of each motion axis. By detecting a given limit position, every drive returns an error frame designated by an identifier of 1B9.

- Being at one or at the other limit switch (state 2 or 3), the maximum excursion of each actuator is scanned to determine the mid-point travel for the platform positioning (state 5). In this state, a CAN frame is sent in writing mode to force the initialization of the position value. Henceforth, the neutral position will be considered as the new reference.
- Once the platform is positioned at the neutral position, an UDP (Universal Datagram Protocol) request with identifier ID: 1 is sent to establish a direct connection with the visual rendering computer. If it responds with an ID: 3, the connection is established (state 6), otherwise a second request is returned. From that moment, the driving simulation can start.
- If an error frame with an identifier of 1B9: XX is detected, a corresponding action must be taken. This frame error is the result of an event more or less critical. Indeed, during the simulation, several unexpected malfunctions may occur. These may be, limit switch reaching, current peak consumption and, overheating. Depending on the severity of the error event, a resolution procedure may be implemented. For example, if the movement has reached a given limit switch, re-scheduling is introduced to stop the simulation smoothly and hence ensuring the simulator's user safety. In the worst case, the drives are switched to control-off and an UDP request with ID: 9 allows the disconnection from the visual environment.

4.8. Motion planning and control

Generation of reference trajectories requires a planning of a motion profile compatible with the mechanics of the simulator and the capabilities of its actuators. In this context, reproducing the full scale vehicle dynamics is impossible even with the more recent mechatronic systems. Inertial forces present in a real driving situation, vary from one vehicle to another and are characterized mainly by accelerations'bandwidth. Designing a system which is able to simulate the frequency content of these accelerations goes inevitably through two different modules: one for rendering high frequency accelerations (also called transient) and the second for low frequency acceleration (also called sustained). For the transient components, a vibration table is an efficient way to reproduce the vehicle linear speed and the road irregularities. The rendering of sustained acceleration is highly dependent on the platform mechanics and on the simulator objectives. If the simulator aims to carry out a behavior study in normal traffic, some experts believe that for manoeuvering below 0.3g (1g= 9.81 m/s^2), a simulator fixed base is sufficient. Otherwise, to simulate a dangerous

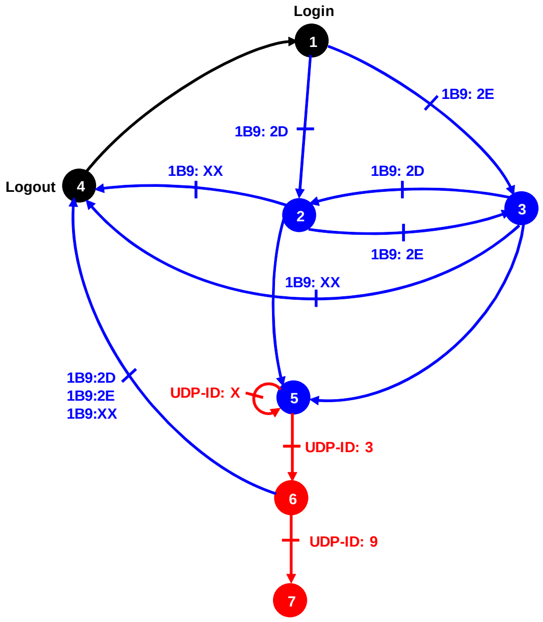

FIGURE 4.7.2. State machine for the simulator monitoring and fault diagnosis

or critical driving situation, the inertial forces restitution is a crucial element, and therefore, a mobile platform is necessary. In the latter case, the first challenge is to accomplish the platform movement in the allowable simulator's workspace. For this, geometric and kinematics constraints should be accounted for and the use of specific planning algorithms commonly known as Motion Cueing Algorithm (MCA) is essential.

MCA has the task of reproducing a part of the inertial forces present in a real driving situation to achieve a sufficient perception level of the simulation [**18**]. These algorithms are based on a simple frequency separation of the various accelerations to be restituted through two principle strategies. The first approach uses the longitudinal motion of the simulator to directly restore the transient acceleration. Moreover, the second one consists on tilting the rider on the mobile platform to take advantage of the gravity vector in order to simulate the sustained component of the linear acceleration. In conclusion, when designing a system for motion cueing, it is important to take into consideration the targeted tests and manipulations. If a vibrating table is sufficient, or must have a large linear motion. Is it better to have no movement than to have a bad one.

However, the planned trajectory, as obtained with the motion cueing algorithm cannot be directly used to actuate the mechanical platform. According to the mechanical architecture of the simulator's platform, it is necessary to transform these trajectories, generated in the Cartesian coordinate space in trajectories expressed in actuators operational space or joint space. It is the role of geometric/kinematic models widely developed in robotics theory [**19**]. In the case of the present simulator, these amounts are used to calculate the jack travels and the displacement of the rear slide according to the desired orientation of the mobile platform.

Once the reference joint trajectories are defined, it is essential to develop a control scheme for the tracking task to drive the mechanical platform and imparting desired transient and steady-state performance. In general, electromechanical systems exhibit a dynamic behavior similar to that of a second order model and therefore can be reasonably controlled by PID controllers (Proportional-Integral-Derivative). Nevertheless, several control schemes intended for complex robotic manipulators like the computed torque method, robust control and sliding mode control, are known in the literature. Since the objective is to achieve a position tracking, all these commands need to implement an external control loop. This solution creates additional time delay in the overall simulation loop and imposes other constraints, especially if commercial drives are adopted.

However, the primary purpose of driving simulation is the creation of an appropriate illusion to lure the simulator user. The exact reproduction of the real movement is of secondary priority, thus, the controller synthesis must move in that direction. Simple strategies favoring compensation of inertial delays have shown their effectiveness. Nowadays,

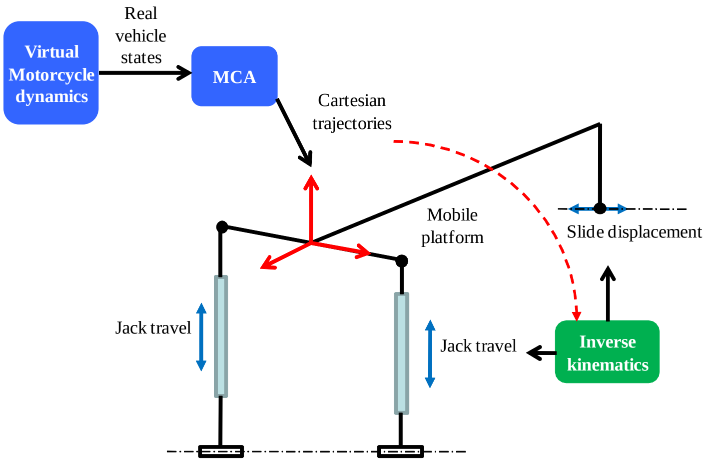

FIGURE 4.8.1. Motion planning by using MCA and inverse kinematics

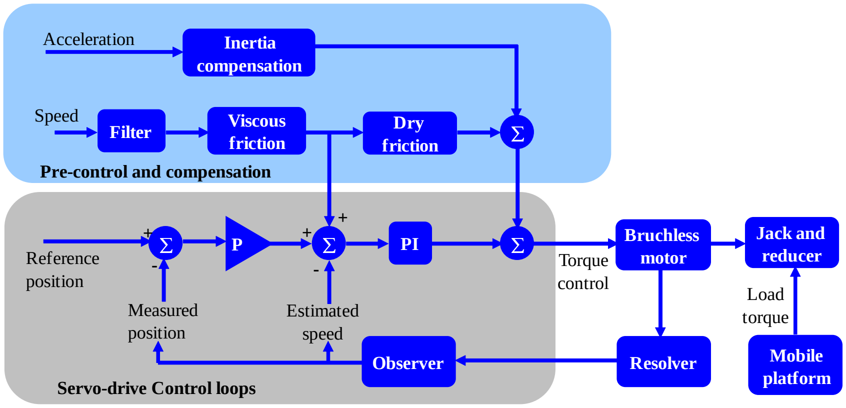

FIGURE 4.8.2. Servo-drive embedded control loops

the servo power monitoring devices are featured with a sufficient intelligence based on axis by axis control with several nested control loops (position, speed and torque). The use of this solution provides significant time savings with a satisfactory performance level.

On the basis of a PID approach, the final control scheme consists of three nested control loops namely, the position loop, the speed loop and the current control loop. The position and velocity are estimated by an observer, from the resolver information of the brushless motor. These estimates are considered as measure variables and are used to compute the tracking error. However, to improve the performance of the servo system, compensation terms of speed and acceleration are expected. These terms, introduced in open loop, do not affect the stability are intended to offset the effect of friction and inertia.

The following diagram shows a commonly PID approach implemented in situ of a commercial drive. In this figure, the current loop does not appear. Indeed, this loop is intended for the motor current tracking according to the reference speed (PI block output) and hence to setup the motor input voltage. This voltage is then chopped and modulated by an appropriate switching scheme for commutating power semiconductor devices and ensures proper monitoring of torque. In a direct current brush motor, the generated torque is generally proportional to the armature current. However, in a three phase PMSM, the torque depends on the phase current amplitude and frequency and therefore the switching algorithm should be very complicated. That is why these motors come with associated drives where the switching and the current loop is already implemented and pre-tuned.

Otherwise, controlling PM brushless DC motor is quite simpler than its counterpart PMSM for low performance applications. The design of a current controller is straightforward, due to the great similarity with DC machines, i.e., PM brushless DC can be modeled as three symmetric DC motors and the inner current control loop can be easily tuned by using a standard linear control approach.

Tuning the different control parameters can be achieved in several ways. As a demonstration, for a position control method, the controller can be expressed by the following equation:

$$\tau_c = k_p e_p + \left(k_v + \frac{k_i}{s}\right)(e_v + k_{fvv}\dot{x}_{ref}) + k_{fa}\ddot{x}_{ref} + k_{fvs}\text{sgn}(\dot{x}_{ref})$$

where, τ_c is the control torque, e_p and e_v are position and speed tracking errors, x_{ref} is the reference position trajectory, k_p, k_v and k_i are the PID controller gains, k_{fvv}, k_{fvs} and k_{fva} are parameters related to the feed-forward compensation terms for viscous and dry friction and inertia, respectively. Next, the dynamic equations of each axis motion must be formulated. To simplify this demonstration, the model of electric motor is not considered because the electrical dynamic is much faster than the mechanical one; therefore, we can write the equation of motion of the motor- jack-reducer as:

$$J_{eq}\ddot{x}_v + f_{eq}\dot{x}_v + f_s\text{sgn}(\dot{x}_v) = N\tau_c - \tau_l$$

where J_{eq}, f_{eq} and f_s are respectively inertia, viscous and dry friction of the equivalent assembly, N is the reducer ratio and τ_l is load torque applied by the simulator mobile platform on the motion axis (like jacks). The load torque can be found from the mobile platform dynamics expressed, by using a simple Newton-Euler modeling approach, by the following equation:

$$\mathcal{M}\ddot{\bar{X}} + \bar{C} + \bar{G} = \mathcal{J}_{-1}^{T}\tau_l$$

Here, $\mathcal{M}$, $\bar{C}$ and $\bar{G}$ are mass matrix, vector of non linearity and related gravity terms vector, respectively. $\mathcal{J}_{-1}^{T}$ is the transpose of the inverse Jacobian matrix allowing the transformation between Cartesian velocities and joint space velocities. Finally, $\bar{X}$ is the vector of the joint coordinates, namely jacks and the rear slide displacements.

These equations can be combined and processed in order to tune the respective parameters of the final controller loops. Nowadays, with the rise of numerical optimization methods, it is possible to introduce constraints for performance and disturbance rejection, while imposing a defined structure of the desired controller. Thus, optimized gains can be used to refine the performance of the inner low-level servo controller and avoiding the need for external control loops. Consequently, platform mechanics is actuated directly by sending reference position trajectories to the motor drives via CAN bus.

4.9. Visual, sound and traffic systems

A visual system consists of a 3D graphics generator and projection devices. Generally, images are projected onto one or more curved or flat screens to ensure a front, side and (if necessary) back vision. In fact, driving is primarily a visual task and it is therefore evident that this information must be carefully produced. Indeed, the image quality is measured by its energy properties (brightness, contrast, resolution, color), its spatial (vision fields and depth) and temporal characteristics (transport delay and refresh rate). Therefore, producing a visual scene with great realism depends on the efforts made to satisfy these factors.

On the other hand, delays are a major problem in applications like driving simulation. Often, these delays are divided into two types: the refresh rate and transportation time. The former is described by the frequency at which the screen is updated or redrawn to give an impression of continuous image animation. The latter, the transportation delay is the time interval between the rider action and the visual projection of the virtual environment. This factor presents a real problem in driving simulation because it is an integral part of the overall simulator mechatronic (virtual motorcycle dynamics, control, acquisition, actuation, image generation, traffic computing, and projection of the visual scene and refresh rate). Singhal and Cheriton [**20**] showed that subjects can detect latencies of 100 (ms), with a tolerance of up 200 (ms). Nevertheless, it was reported that the overall delay in a driving simulator should not exceed 50 (ms) [**21**]. Thus, if the refresh rate is 60Hz (equivalent to 17 ms), it is evident, with a simple subtraction, that all other sub-systems should operate and communicate at delays less than 33 (ms).

Moreover, depriving drivers of acoustic cues leads to a systematic increase in the speed of the vehicle. To prevent this issue, a 3D sound system is used. Indeed, the sound system allows the enhancement of the driving simulation realism. The main characteristics of this element are: the number, quality and location of speakers, and also the diversity of sounds. Indeed, the dominant sound frequency band in a vehicle is about 20 to 500 (Hz), induced mainly by the engine [**22**]. Some sounds from tire/road interaction have major components at high frequencies. Therefore, the sound system should cover a few thousand hertz. Consequently, to achieve a realistic sound illusion, it is paramount to generate a spatial multi-channel sound. If a vehicle passes nearby, the direction of the sound should follow the projected image to avoid the rider disorientation.

Finally, traffic and scenario are important elements as they add some interactivity to the visual environment. Starting from the individual driver's behavior, visual objects like vehicles and pedestrians are imparted with a defined level of intelligence to immerse the simulator's user in realistic traffic conditions.

4.10. Rider safety versus existing security systems

The automotive industry has greatly evolved in recent years. Today, cars are considered as complex mechatronic systems. Each feature is seen with a dedicated processor for managing multiple sensors and processing a considerable flow of data. This complexity is the result of a massive spreading of security features, support, comfort and power management. In particular, driving safety has become the unifying theme among car manufacturers, official institutions and customers. The present challenge consists in developing security assistance and support, more efficiently and less costly.

In this context, the number of deaths in road accidents has experienced a huge reduction (about 50%) during the last decade. However, analysis of accident statistics shows that the number of deaths when a motorcycle is involved, has increased. In fact, the motorcycle remains a particularly dangerous mode of transport: the number of deaths is still very high, and if one takes account of the number of traveled kilometers, the risk of death for a motorcycle rider is 21 times higher than that of other transportation modes. In additions, motorcycles and scooters are becoming more and more popular for urban an suburban travel in European countries. The associated traveled kilometers and the number of vehicles sold is continuously on the rise.

During the last *twenty* years, passenger cars experienced several advances in passive and active safety systems. Nowadays, cars are all equipped with one or several airbags and ABS (Anti-lock Braking System) systems. More powerful systems such as ESP (Electronic Stability System), brake assist systems and traction control systems or belt tensioners are becoming increasingly widespread. Conversely, during the same period, the backwardness taken by motorcycles continues to grow. For example, ABS, which has been around for over 15 years, is still reserved to few top range motorcycle models. The braking distributor is becoming less expensive but still remains marginal in the world of motorcycles. The use of motorcycle airbags, whose development seems problematic, did not become widespread. The recent braking amplifier seems to be promising, but its diffusion remains, very restricted.

4.11. Conclusion

The design of each driving simulator is a very difficult multidisciplinary challenge. Each component is essential (vision, motion restitution, communication, ...etc.) for the success of the driving simulation and to increase the immersion level. One of these important aspects is mechatronics, including mechanics, embedded electronics, actuation technologies and control laws, all in a single system).

The design of a driving simulator is based, among others, on a mechanical platform and an embedded electronics for its achievement. The performances of the final architecture are checked by a juxtaposition of the different components. Therefore, an optimal design of such systems requires a multi-physic modeling and optimization of the global system.

The quality of a driving simulator is rated by its fidelity level. The quality of motion restitution is largely dependent on the platform's mechatronics. Without this condition, perceptive errors are transmitted to users and can make them quite uncomfortable. Thus, as stated before, an optimized mechatronics strongly helps the system to provide optimum motions.

Bibliography

[1] Dagdelen M, Reymond G, Kemeny A, Bordier M, Maiza N. MPC based motion cueing algorithm : Development and application to the ULTIMATE driving simulator. Driving Simulation Conference DSC Europe 2004; 221-233.

[2] Nehaoua L, Mohellebi H, Amouri A, Arioui H, Espié S, Kheddar A. Design and control of a small-clearance driving simulator. IEEE Transactions on Vehicular Technology 2008 ; 57(1) :736-746.

[3] Arioui H, Hima S, Nehaoua L, Bertin RJV, Espié S. From Design to Experiments of a 2-DOF Vehicle Driving Simulator. IEEE Transactions on Vehicular Technology 2011 ; 60(2) :357-368.

[4] Miyamaru Y, Yamasaky G, Aoky K. Development of motorcycle riding simulator and its prehistory. JSME Review 2000; 50.

[5] Yamasaky G, Aoky K, Miyamaru Y, Ohnuma K. Development of motorcycle training simulator. JSAE Review 1998; 19: 81-85.

[6] Ferrazzin D, Barbagli F, Avizzano CA, Pietro DD, Bergamasco M, Designing new commercial motorcycles through a highly reconfigurable virtual reality-based simulator. Advanced Robotics 2003; 17(4): 293-318.

[7] Kwon DS. Kaist interactive bicycle simulator. Proceedings of IEEE International Conference on Robotics and Automation (ICRA01) ; 2313-2318.

[8] Cossalter V, Doria A, Lot R. Development and validation of a motorcycle riding simulator. World Automotive Congress FISITA 2004, Barcelona

[9] Arioui H, Nehaoua L, Hima S, Séguy N, Espié S. Mechatronics, design and modeling of a motorcycle riding simulator. IEEE/ASME Transactions on Mechatronics 2010; 15(5): 805-818.

[10] Pacejka HB, Sharp RS. Shear force development by pneumatic tyres in steady state conditions : A review of modeling aspects. Vehicle System Dynamics 1991; 20 :121-176.

[11] Cossalter V, Lot R. A motorcycle multibody model for real time simulation based on the natural coordinates approach. Vehicle System Dynamics 2002 ; 37(6) :423-447.

[12] Khalil W, Dombre E. Modélisation, identification et commande des robots. Hermes science publications, Paris, 2nd edition,1999.

[13] Hollerbach JM. A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. IEEE Transaction on systems, man and cybernetics 1980; 10(11): 730-736.

[14] Hima S, Nehaoua L, Séguy N, Arioui H. Suitable two wheeled vehicle dynamics synthesis for interactive motorcycle simulator. Proceedings of the 17th IFAC World Congress 2008; 96-101.

[15] Sharp RS. The stability and control of motorcycles. Journal of Mechanical Engineering Science 1971; 13:316-329.

[16] Bishop RH. Mechatronic systems, sensors, and actuators: fundamentals and modeling. CRC Press Inc 2007.

[17] xPC Target for use with real-time workshop. User's guide. The MathWorks.

[18] Nahon MA, Reid LD. Simulator motion-drive algorithms : A designers perspective. Journal of Aircraft 1990; 13(2): 356-362.

[19] Nehaoua L, Hima S, Arioui H, Séguy N, Espié S. Design and modeling of a new motorcycle riding simulator. Proceedings of the 2007 American Control Conference. 176-181.

[20] Singhal SK, Cheriton DR. Exploiting position history for efficient remote rendering in networked virtual reality. Teleoperation and Virtual Environment, Presence 1995; 4(2) :169-19.

[21] Franck LH, Casalli JG, Wierville WW. Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator. Human Factors 1988; 30 :201-217.

[22] Genuit K, Bray W. A virtual car : Prediction of sound and vibration in an interactive simulation environment. Proceedings of SAE Noise & Vibration Conference & Exposition, Grand Traverse 2001, USA.

[23] Larnaudie B, Bouaziz S, Maurin T, Espié S, Reynaud R. Motorcycle platform for dynamics model extraction. Proceedings of the IEEE Intelligent Vehicle Symposium 2006.