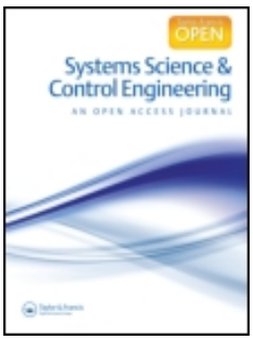

Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) 2164-2583 (Online) Journal homepage: https://www.tandfonline.com/loi/tssc20

Effective fault-tolerant control paradigm for path tracking in autonomous vehicles

Afef Fekih & Shankar Seelem

To cite this article: Afef Fekih & Shankar Seelem (2015) Effective fault-tolerant control paradigm for path tracking in autonomous vehicles, Systems Science & Control Engineering, 3:1, 177-188, DOI: 10.1080/21642583.2014.1002138

To link to this article: https://doi.org/10.1080/21642583.2014.1002138

Systems Science & Control Engineering: An Open Access Journal, 2015
Vol. 3, 177–188, http://dx.doi.org/10.1080/21642583.2014.1002138

Taylor & Francis
Taylor & Francis Group

Effective fault-tolerant control paradigm for path tracking in autonomous vehicles

Afef Fekih* and Shankar Seelem

Department of Electrical and Computer Engineering, University of Louisiana at Lafayette, Lafayette, LA, USA

(*Received 21 July 2014; accepted 21 December 2014*)

A novel fault-tolerant control paradigm that integrates fault detection (FD) with optimal control for path tracking is designed to ensure accurate path tracking in the presence of faults. The proposed approach is designed to maintain vehicle stability, dynamics, and maneuverability in the event of a faulty steering system. A sensor fusion-based fault detection and identification approach is proposed to accurately detect and identify sensor faults when they occur. A weight adjustment algorithm is considered to ensure accurate detection while providing robustness against parameter variations and uncertainties. Following FD and using the estimated fault vector, a fault-tolerant controller is designed to guarantee the stability of the closed loop system. The proposed controller incorporates a linear quadratic regulator (LQR)-based algorithm with a feed-forward gain. The LQR-based controller is designed to maintain system stability under faulty conditions while operating the dynamic system at minimum cost. The proposed approach is validated using a ground vehicle required to track various paths while being subject to multiple fault scenarios. For accurate performance analysis, vehicle handling and dynamics were implemented using CarSim, a high-fidelity vehicle simulator. Effective path tracking capabilities, vehicle handling, and stabilization under both fault-free and faulty conditions are the main positive features of the proposed approach.

Keywords: fault detection and identification (FDI); fault-tolerant control (FTC); linear quadratic regulator (LQR); sensor fusion; observer

1. Introduction

As modern day autonomous vehicles become more and more complex and highly integrated, the vulnerability of their components to faults/failures increases (Cheng, 2011; Fekih, 2014). Defects in sensors, actuators, or the system itself can degrade overall system performance. Undetected, faults can develop into failures which probably increase with the increased complexity of the system. Moreover, mitigating unsatisfactory performances or even instability caused by the unpredictable faults in actuators, sensors, or other components is of foremost priority, especially in safety-critical systems such as ground vehicles.

According to the International Federation of Automatic Control SAFEPROCESS technical committee (Gustaffson, 2000), a fault is defined as any unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition, while a failure is a permanent interruption of a system's ability to perform its function under specific operating conditions. In order to maintain high levels of performance and guarantee proper system behavior, it is important that faults be promptly detected and identified and appropriate remedies be applied to prevent system malfunctions. Diagnosis is the primary stage of active fault-tolerant control (FTC) systems. Its goal is to perform two main decision tasks: fault detection (FD), consisting of deciding whether or not a fault has occurred, and fault isolation, consisting of deciding which element of the system has failed (Fekih, 2014).

A FTC system is a control system specifically designed to automatically accommodate faults among system components while maintaining system stability along with a desired level of overall performance (Blanke & Staroswiecki, 2006; Noura, Theilliol, Ponsart, & Noura, 2009). The key issue of a FTC system is to prevent local faults from developing into system failures that can end the mission of the system and cause safety hazards for man and environment. Existing efforts in FTC design can be classified into two main approaches: the passive and active approaches (Jiang & Yu, 2012). In the passive approach, robust control techniques are used to ensure that the control loop system remains insensitive to certain faults. The effectiveness of this strategy, which usually assumes very restrictive repertory of faults, depends upon the robustness of the nominal closed-loop system. In the active approach, a new control system is redesigned according to the estimation of the fault performed by the fault detection and identification (FDI) filter and according to the specifications to be met by the faulty system. In contrast to passive approaches that are mostly conservative, active approaches

*Corresponding author. Email: afef.fekih@louisiana.edu

are able to deal with a large number of fault scenarios and can handle a certain number of unforeseen faults that were not considered at the design stage.

A growing body of research in this area has resulted in a number of FD and FTC schemes for ground vehicles (Arogeti, Wang, Low, & Yu, 2008; Chen, Song, & Li, 2011; Dong, Verhaegen, & Holweg, 2008; Fekih & Seelem, 2012; Herpin, Fekih, Golconda, & Lakhotia, 2007; Laureiro, Benmoussa, Touati, Merzouki, & OuldBouamama, 2014; Morteza & Fekih, 2014a, 2014b; Tabbache, Benbouzid, Kheloui, & Bourgeot, 2011; Wang & Wang, 2013; Yang, Cocquempot, & Jiang, 2008). A passive actuator FTC was proposed for Four-Wheel Independently Actuated electric ground vehicle in Wang and Wang (2013). The approach exploits the redundancy of the system and groups actuators with similar faults in one subsystem and applies control allocation to distribute the control effort. Actuator grouping was attempted to reduce the significant computational cost typically associated with control allocation. In Laureiro et al. (2014) a bond graph model-based FD approach and an FTC were designed for an over-actuated heavy size autonomous vehicle. The approach relied heavily on analytical redundancy relations derived from the bond graph model. A robust and adaptive FTC tracking approach was proposed in Chen et al. (2011). An FTC strategy which considers a maximum-likelihood voting algorithm was proposed for sensor faults in Tabbache et al. (2011). Note that most of these works exploited system redundancy and required high computational costs, drawbacks that might prevent their real-time implementation.

In this paper, an FTC framework that implicitly integrates FDI with FTC is designed for the automatic steering of an autonomous ground vehicle subject to sensor faults. The proposed controller is based on a linear quadratic regulator (LQR) augmented with a feed-forward gain. The LQR-based controller is designed to place the system's eigenvalues in the stable region while operating the dynamic system at minimum cost function. An observer-based FDI approach is proposed to detect and identify sensor faults when they occur. Using the estimated fault vector, the fault-tolerant controller is designed to maintain system stability when faults occur. The proposed framework is implemented on a ground vehicle required to follow a given path, while being subject to sensor faults. The steering controller is designed to maintain vehicle stability, dynamics, and maneuvrability in the event of a faulty steering system.

Compared with the existing work already reported in the literature (Arogeti et al., 2008; Chen et al., 2011; Laureiro et al., 2014; Morteza & Fekih, 2014b; Tabbache et al., 2011; Wang & Wang, 2013), the contributions of this paper are in the following aspects:

It presents a complete FTC design with the FDI algorithm as integral part of the framework and applies it to the automatic steering of an autonomous vehicle.

The FDI algorithm incorporates a weight adjustment algorithm to ensure accurate detection, while providing robustness against parameter variations and uncertainties.

It integrates the optimal properties of the LQR framework with an observer-based fault detection scheme to achieve effective fault tolerance.

It provides an easy to implement algorithm which achieves fault-tolerance with optimum computational costs. This is crucial in autonomous vehicles, which often work under tight real-time deadlines and cannot tolerate prolonged delays in control reconfiguration.

The rest of this paper is organized as follows. Section 2 presents the dynamic model of the vehicle and discusses the design specifications. The proposed control paradigm is detailed in Section 3. Section 4 is dedicated to the performance analysis of the proposed algorithm. Finally, some concluding remarks end this paper in Section 5.

2. Vehicle dynamic model and problem formulation

A dynamic model of the vehicle, with the front and rear wheels lumped together into a pair of single wheels at the center of gravity (CG) (Fekih & Deveriste, 2013), is considered as shown in Figure 1.

Assuming constant longitudinal velocity and combining the lateral forces with the available slip angles, the vehicle's dynamic model is defined as follows (Fekih & Deveriste, 2013):

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{-(c_f + c_r)}{m v_x} & \dfrac{l_r c_r - l_f c_f}{m v_x} - v_x \\ \dfrac{l_r c_r - l_f c_f}{I_z v_x} & \dfrac{-(l_f^2 c_f + l_r^2)}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \dfrac{c_f}{m} \\ \dfrac{l_f c_f}{m} \end{bmatrix} \delta, \tag{1}$$

where $\dot{v}_y$ is the rate of change of lateral velocity, $\dot{r}$ is the yaw rate of the vehicle, δ is the steering angle, θ is the yaw

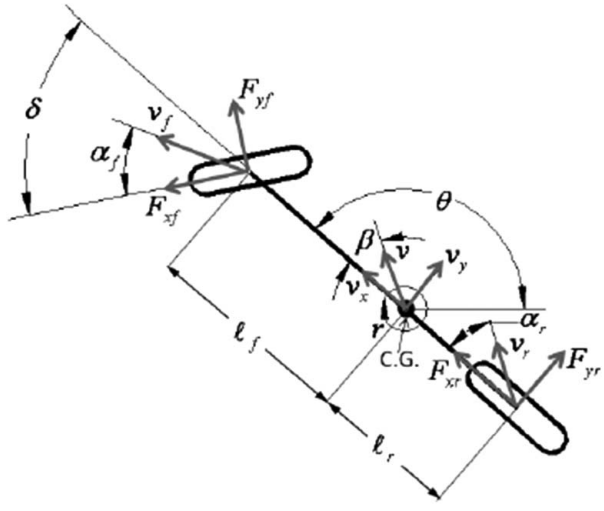

Figure 1. Dynamic bicycle model.

angle (orientation angle of the vehicle with respect to the X axis), and v_y, v_x are the lateral and longitudinal velocity, respectively. c_f and c_r are the cornering stiffness of the front and rear tires, respectively. ℓ_f is the distance from the CG to the front axle and ℓ_r is the distance from the CG to the rear axle. I_z is the vehicle yaw moment of inertia. The remaining variables and parameters are defined in the Appendix (Table A1). Since our objective is to develop a steering control system for automatic lane keeping, the state variables are being expressed in terms of position and orientation error. If we consider a vehicle traveling at a constant velocity on a road of a large radius with curvature k and assume a constant longitudinal velocity, the rate of change of the desired orientation of the vehicle is given by

$$\dot{r}_{\text{des}} = v_x k, \tag{2}$$

where $\dot{r}_{\text{des}}$ is the desired yaw rate and k is the curvature of the road. The desired path lateral acceleration of the vehicle can be written as

$$\dot{v}_y = k v_x^2. \tag{3}$$

Define e as the distance of the CG of the vehicle from the center line of the path and e_1 as the yaw angle error of the vehicle with respect to the path; then, we have

$$\begin{aligned} \ddot{e} &= \ddot{v}_y + v_x(\dot{r} - \dot{r}_{\text{des}}), \\ \dot{e} &= \dot{v}_y + v_x(r - r_{\text{des}}), \\ e_1 &= r - r_{\text{des}}. \end{aligned} \tag{4}$$

The state-space model in tracking error variables is therefore given by

$$\dot{x} = Ax + B_1\delta + B_2\dot{r}_{\text{des}}, \tag{5}$$

with $x = [e \quad \dot{e} \quad e_1 \quad \dot{e}_1]^{\text{T}}$.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\dfrac{(c_f + c_r)}{m v_x} & \dfrac{c_f + c_r}{m} & \dfrac{l_f c_f - l_r c_r}{m v_x} \\ 0 & 0 & 0 & 1 \\ 1 & \dfrac{l_r c_r - l_f c_f}{I_z v_x} & \dfrac{l_f c_f - l_r c_r}{I_z} & -\dfrac{(l_f^2 c_f + l_r^2 c_r)}{I_z v_x} \end{bmatrix} \ldots,$$

$$B_1 = \begin{bmatrix} 0 \\ \dfrac{c_f}{m} \\ 0 \\ \dfrac{l_f c_f}{m} \end{bmatrix}, \ldots B_2 = \begin{bmatrix} 0 \\ \dfrac{l_r c_r - l_f c_f}{m v_x} - v_x \\ 0 \\ -\dfrac{(l_f^2 c_f + l_r^2 c_r)}{I_z v_x} \end{bmatrix}.$$

The output vector of the system consists of measurements from the two sets of sensors. Sensor failures are modeled as additive signals to the sensor output as follows:

$$y = [y_1 \quad y_2]^{\text{T}} = Cx + Ff, \tag{6}$$

where y_1 and y_2 are the measurements from the two sensors which measure the lateral deviation of the vehicle and f is the fault signal, which is a function of time and state x, and C is the output matrix, defined as $C = \left[\begin{smallmatrix} C_1 \\ C_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 1 & 0 & d_1 & 0 \\ 1 & 0 & -d_2 & 0 \end{smallmatrix}\right]$, where d_1 and d_2 are, respectively, the distances from the CG of the vehicle to the front and rear bumpers where the sensors are located. $F = [1 \quad 0]^{\text{T}}$ in the event of failure of sensor 1 and $F = [0 \quad 1]^{\text{T}}$ in the event of failure of sensor 2, respectively. Here, we consider that the lateral sensing system consists of two sets of sensors which provide the information of the lateral deviation.

3. FTC paradigm

The faults under investigation are sensor faults with varying severities and types. The FTC objectives are to maintain vehicle stability, dynamics, and maneuverability in the event of faulty sensors.

3.1. *Observer-based FDI algorithm*

The following are some important features of the vehicle's dynamic model that can aid in designing an easy-to-implement FDI algorithm: (1) it has two zero eigenvalues and (2) (A, C_1) and (A, C_2) are observable. This implies that we can estimate the state through either y_1 or y_2. This makes FDI easy to implement with minimum computational cost.

The observability properties of the vehicle imply that we can build two observers, each of which is being driven by a single sensor output. Furthermore, in order to ensure that no erroneous estimates of the state are obtained under sensor failures, we fuse the sensor output and the estimated output from one observer prior to their use by the other observer. Fusion blocks play the role of switches, which select the healthy signal. Then post-filters are designed such that the transfer functions from fault signals to residuals have consistent behavior in order to facilitate fault identification.

Output fusion is a convex combination of the sensor output and the estimated output from the observer. The fused output y_{fi} is given by

$$y_{fi} = (1 - \lambda_i) y_i + \lambda_i \hat{y}_i^j, \tag{7}$$

where $\hat{y}_i^j$ is the estimate of the ith output from the jth observer, and $i, j = 1, 2$, and λ_i is a weight function. The weights $\lambda_i \in [0, 1]$ are adjusted using a weight adjustment algorithm. When there are no faults, that is, $\lambda_i = 0$, then the fused output y_{fi} is identical to the sensor output y_i. When faults occur, the corresponding values of λ_i will increase toward one. When $\lambda_i = 1$, the sensor output is

incorrect and therefore is not being taken into account at all. The observers will switch between two configurations according to the relative size of the weights as follows:

If $\lambda_1 < \lambda_2$, then the observers are defined by

$$\begin{aligned}\dot{\hat{x}}_1 &= A\hat{x}_1 + B_1\delta + L_1(y_{f\,1} - \hat{y}_1^1) + \lambda_1 L_1 C_1(\hat{x}_1 - \hat{x}_2),\\ \dot{\hat{x}}_2 &= A\hat{x}_2 + B_1\delta + L_2(y_{f\,2} - \hat{y}_2^2).\end{aligned} \tag{8}$$

If $\lambda_1 > \lambda_2$, then

$$\begin{aligned}\dot{\hat{x}}_1 &= A\hat{x}_1 + B_1\delta + L_1(y_{f\,1} - \hat{y}_1^1)\\ \dot{\hat{x}}_2 &= A\hat{x}_2 + B_1\delta + L_2(y_{f\,2} - \hat{y}_2^2) + \lambda_2 L_2 C_2(\hat{x}_2 - \hat{x}_1),\end{aligned} \tag{9}$$

where L_1 and L_2 are the solutions of the characteristic equation defined by $\det(sI\text{-}A + L_1C_1)$ and $\det(sI\text{-}A + L_2C_2)$, respectively. Observers (8)–(9) are variations of the Luenberger observer where the fused outputs replace the sensor outputs.

3.2. Threshold generation logic

For accurate FD, define the following fault indicator function or threshold logic:

$$\begin{aligned}\|r_i(t)\| < T_{\mathrm{h}} &\Rightarrow \text{fault-free conditions},\\ \|r_i(t)\| > T_{\mathrm{h}} &\Rightarrow \text{faulty conditions},\end{aligned} \tag{10}$$

where T_{h} is a predefined threshold typically chosen based on the application at hand. Note that setting low thresholds results in high false-positive rates (alarms are issued under no fault conditions), and setting high thresholds increases the false-negative rates (alarms are missed when faults occur). Clearly, the selection of the thresholds is closely related to robustness and sensitivity of the residual generator. Different analysis procedures are used depending on the techniques employed to generate the residual signal. The most widely used approaches to analyze the residual signal generated by observers are threshold logic and limit monitoring. Threshold level selection methods are generally problem specific and are not useful for a general case (Hsiao & Tomizuka, 2005). To avoid improper FD, threshold level selection is often done on the basis of the designer's experience and in response to problem requirements.

3.3. Post-filters

Consider the error vector:

$$e_{yi} = \begin{bmatrix} y_1 - \hat{y}_1^1 & y_1 - \hat{y}_1^2 & y_2 - \hat{y}_2^1 & y_2 - \hat{y}_2^2 \end{bmatrix}^{\mathrm{T}}. \tag{11}$$

Residuals are generated by filtering e_{y} through post-filters $M_i(s)$, that is,

$$r_i = M_i e_{yi} \quad i = 1, 2, 3, 4. \tag{12}$$

$M_i(s)$ define the transfer functions from the faulty signals to the residuals such that the residuals from the two observers are comparable in magnitude. Note that r_1 and r_2 are related to sensor 1 and r_3 and r_4 are related to sensor 2, respectively. Notice that observers (8)–(9) are coupled, that is, faults in either of the two sensors affect all residuals. The problem of identifying the exact fault sensor can be solved by using properly designed post-filters (Zhang, Ding, Lam, & Wang, 2003).

Define the state-space transfer function from fault f to e_{y} by

$$V_i(s) = C(sI - A)^{-1}B + F, \tag{13}$$

where F is a matrix which represents the sensor failure as follows:

$$\begin{aligned} F &= \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathrm{T}} \quad \text{If sensor 1 fails},\\ F &= \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathrm{T}} \quad \text{If sensor 2 fails}.\end{aligned}$$

Consider the scenarios when sensor 1 has failed and $\lambda_1 < \lambda_2$, then from Equation (12) we have

$$\begin{aligned} V_1(s) &= -(1-\lambda_1)C_1(sI - A + (1-\lambda_1)L_1C_1)^{-1}L_1 + 1\\ &= \frac{n_1(s)}{\mathrm{d}(s)}, \end{aligned} \tag{14}$$

$$\begin{aligned} V_3(s) &= -(1-\lambda_1)C_2(sI - A + (1-\lambda_1)L_1C_1)^{-1}L_1\\ &= \frac{(1-\lambda_1)n_3(s)}{\mathrm{d}(s)}, \end{aligned} \tag{15}$$

where $((n_1(s), d(s))$ and $(n_3(s), d(s))$ are the co-prime pairs of the polynomials defined as follows:

$$n_1(s) = \det\left(\begin{bmatrix} sI - A & L_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ (1-\lambda_1)C_1 & 1 \end{bmatrix}\right), \tag{16}$$

$$n_3(s) = \det\left(\begin{bmatrix} sI - A & L_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ (1-\lambda_1 C_2 & 1 \end{bmatrix}\right), \tag{17}$$

where $n_1(s)$ and $n_3(s)$ are also independent of λ_1.

Factorizing $n_1(s) = n_1^+(s)n_1^-(s)$ and $n_3(s) = n_3^+(s)n_3^-(s)$, where $n_i^+(s)$ and $n_i^-(s)$, $i = 1,3$, are the factors of $n_i(s)$ which have their roots in the left half plane.

Choosing:

$$M_1(s) = \frac{{n_1}^+(s)}{n_3^-(s)k(s)}, \tag{18}$$

$$M_3(s) = \frac{n_3^+(s)}{n_1^-(s)k(s)}, \tag{19}$$

where $k(s)$ is a Hurwitz polynomial such that $M_1(s)$ and $M_3(s)$ are proper and stable, yields

$$(1-\lambda_1)M_1(s)V_1(s) = M_3(s)V_3(s). \tag{20}$$

This implies that if we choose post-filters M_{i}'s such that $a_1M_1V_1 = M_3V_3$ and $a_2M_4V_4 = M_2V_2$ for some real

numbers $a_1 > 0$ and $a_2 < 1$, we can define the following identification rules:

If $\lambda_1 < \lambda_2$, and a fault was detected, $|r_1| > |r_3|$ indicates that sensor 1 has failed, while $|r_1| < |r_3|$ implies that sensor 2 has failed.

Similarly, if $\lambda_1 > \lambda_2$, and a fault has been detected, $|r_2| > |r_4|$ suggests that sensor 1 has failed, while $|r_2| < |r_4|$ implies that sensor 2 has failed. In order to accommodate faults, when these latter happen, the controller considers the fused outputs, that is, y_{f1} and y_{f2} instead of the sensor outputs y_1 and y_2.

If the fault occurs, the faulty sensor output is replaced by the observer output.

The next step after residual generation is the analysis of the residual signal for FD. The residual generator takes the sensor measurements as inputs and generates residuals. The latter are small, ideally zero, when there are no fault; but when a fault occurs, the residuals are significantly large. Due to the effect of disturbances, model uncertainties, and measurement noise, the residuals are different from zero even when there are no faults. A robust residual generator is proposed next to alleviate these effects while remaining sensitive to faults.

3.4. Weight adjustment algorithm

If any fault has been detected and identified, weights λ_i, $i = 1, 2$ in the fusion blocks will be adjusted. Suppose sensor 1 fails, then we adjust the weights according to the following 1st order differential equations:

$$\dot{\lambda}_1 = -\alpha(\lambda_1 - g(|[r_1 \quad r_2]^{\mathrm{T}}|) \tag{21}$$

$$\dot{\lambda}_2 = -\alpha\lambda_2. \tag{22}$$

If sensor 2 fails, then the adaption rule becomes

$$\dot{\lambda}_1 = -\alpha\lambda_1 \tag{23}$$

$$\dot{\lambda}_2 = -\alpha(\lambda_2 - g(|[r_3 \quad r_4]^{\mathrm{T}}|), \tag{24}$$

where g is a sigmoid function defined as follows: g: R $\rightarrow$[0, 1] $g(x) = 1/(1 + e^{-ax})$ $a > 0$ with $a \in [-10, 10]$. The value of a is determined based of the fault magnitude. For instance, complete failure of sensor 1 results in $a = -10$, hence the use of information from residuals r_1 and r_2. If no failure is reported, then $a = +10$, resulting in $g = 0$, hence residual signals not being considered in the adjustment algorithm. Values between -10 and 10 represent faults with various magnitudes. The sufficient conditions for convergence of the estimated state are $|\dot{\lambda}_1| < \alpha$ and $\lambda_1 + \lambda_2 \leq 1$. The parameter $\alpha > 0$ and is a trade-off between stability and FDI performance. Large α increases the response of the FDI unit to faults, while small α results in a slowly varying condition which guarantees system stability.

3.5. FTC strategy

When a fault is detected, the state variables are reconstructed accordingly and the feedback controller is redesigned as follows:

$$\delta(t) = -K_f\,\hat{x}(t), \tag{25}$$

where K_f is the feedback controller gain when the lateral control system enters into a degraded mode, that is, when the information is lost in one of the lateral deviation sensors. A fault in any of the sensors results in a change in the output measurements and the state variables are defined as follows:

$$\dot{x}(t) = Ax(t) + B_1\delta(t) + B_2\dot{r}_{\mathrm{des}}, \tag{26}$$

$$y(t) = \bar{C}x(t) + Ff, \tag{27}$$

where A is the state matrix, B_1 is the control matrix, and $\bar{C}$ is the output matrix of the faulty system; $\dot{r}_{\mathrm{des}}$ is the desired yaw rate, f is the additive fault signal, and F represents sensor faults. Now an optimal estimator can easily be designed by solving the relevant Riccati equation associated with the system given by Equations (25) and (26). Assuming the system is observable, the state estimates $\hat{x}$ are defined as follows:

$$\dot{\hat{x}}(t) = (A - L\bar{C})\hat{x}(t) + B_1\delta(t) + Ly(t), \tag{28}$$

where L is the observer gain which is defined by $L = Y\bar{C}B_1^{-1}$ and Y is the positive semi-definite stabilizing solution of the following algebraic Riccati equation:

$$(A - B_1\bar{C})Y + Y(A - \bar{C})^{\mathrm{T}} - Y\bar{C}Y + B_1B_1{}^{\mathrm{T}} = 0. \tag{29}$$

In order to guarantee the stability of the proposed observers, define the estimation error $\varepsilon(t)$ as follows:

$$\varepsilon(t) = x(t) - \hat{x}(t). \tag{30}$$

The error dynamics are stable if and only if the matrix $U = \left[\begin{smallmatrix} A-LC & K \\ -C & 1 \end{smallmatrix}\right]$ is Hurwitz stable.

Note that the matrix U can be written as follows:

$$U = \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} L \\ 1 \end{bmatrix}[C \quad 0]. \tag{31}$$

Therefore, the poles of the matrix U can be arbitrarily assigned, provided that $\left(\left[\begin{smallmatrix} A & B \\ 0 & 1 \end{smallmatrix}\right], [C \quad 0]\right)$ is observable. Hence, stability of the proposed observer is guaranteed by the proper choice of observer gain L, which is selected in a way such that the matrix U is Hurwitz stable.

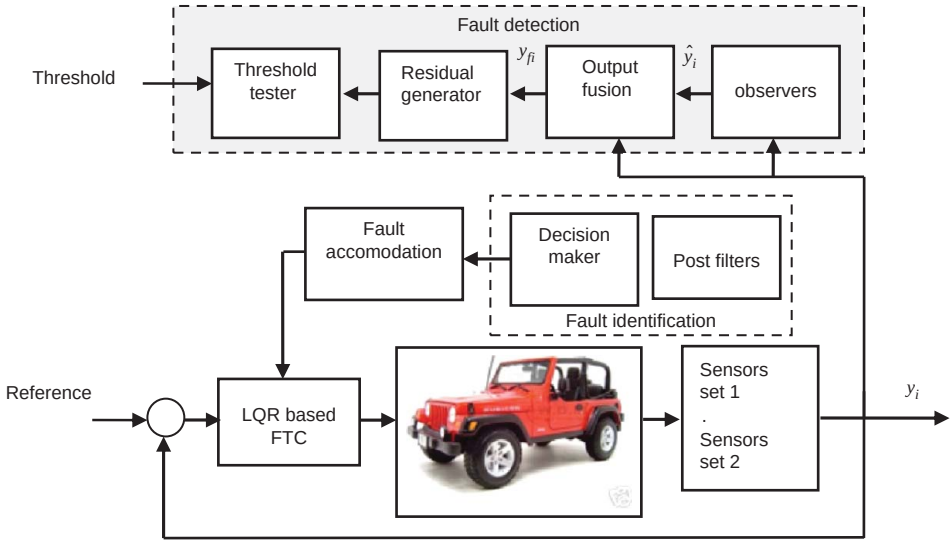

Figure 2. Schematic diagram of the proposed FTC framework.

Define the objective cost function to be minimized by the controller as follows:

$$J = \int_0^\infty \hat{x}^{\mathrm{T}}(t)Q\hat{x}(t) + \dot{\delta}^{\mathrm{T}}(t)R\delta(t), \tag{32}$$

where P satisfies the following Riccati equation:

$$P = A^{\mathrm{T}}PA - A^{\mathrm{T}}PB_1(R + B_1^{\mathrm{T}}PB_1)^{-1}B_1^{\mathrm{T}}PA + Q. \tag{33}$$

Here, Q is a diagonal weighting matrix with an entry for each state corresponding to the performance aspects contributing to the cost function and R is the weighting value corresponding to the control effort contributing to the cost function.

The proposed FTC framework is illustrated in Figure 2.

The approach uses two observers driven by sensor outputs. Following FD, faulty sensors are identified. The state variables are then constructed from the output of the healthy sensor and the controller is updated accordingly to ensure proper steering and maintain system stability when faults occur.

4. Application of the FTC framework to an automotive steering system

To validate the performances of the proposed control paradigm, we provide a series of computer experiments using various paths and fault scenarios. For accurate evaluation, the proposed controller is implemented using CarSim (Mechanical Corporation). The latter is used in the automotive industry as the standard by which vehicle handling and dynamics are tested. It provides a high-fidelity and complete model of the vehicle and its environment. The performance of the proposed control paradigm is compared to that of the CarSim steering controller, which details are illustrated in the Appendix. Two driving maneuvers are chosen to perform the various experiments as detailed in the following.

4.1. *Lane change maneuver*

Lane change maneuver is a common test for vehicle handling as it represents an essential collision avoidance maneuver. A lane change path is chosen to demonstrate the tracking capability on a straight path as well as the response to a quick, yet continuous transient section (position and curvature). Experiments on this path are performed at a constant longitudinal speed of 30 m/s (108 km/h) and considering a road adhesion factor of one.

A double lane change path is selected to illustrate the tracking capability as well as the steering control of the vehicle on a straight path. Figures 3 and 4 show the vehicle following a lane change maneuver. Computer experiments were first carried out when 90% of the information from the sensor is lost and the front sensor has failed at $t = 4$ sec. For comparison purposes, simulations were carried out with and without FTC. The fault considered here is an abrupt change in the front sensor.

Figures 5 and 6 depict the responses of the vehicle at a longitudinal speed of 40 m/s (144 km/h) and considering a road adhesion factor of 0.75 when 90% of the sensor information is lost ($a = 3$). The observer gains in this case were $L_1 = 10$ and $L_2 = 20$, while the controller gain was $K = 80$. Note a reduction in the friction between the road and the tire of the vehicle in this case. We can observe from the figures that when the FTC approach is considered, the vehicle is able to recover from the fault fairly quickly and follow the prescribed path without much delay. This is important since lane change control operation is high bandwidth in nature and cannot tolerate significant delays in the

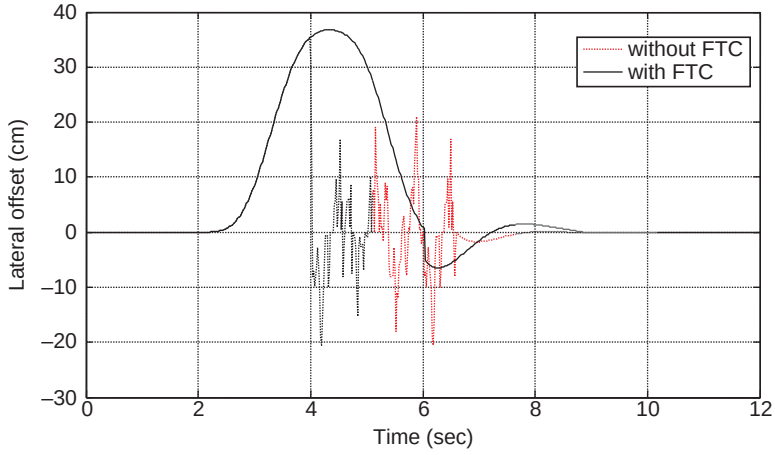

Figure 3. Lateral offset with 90% loss at $v = 30$ m/s and $u = 1$.

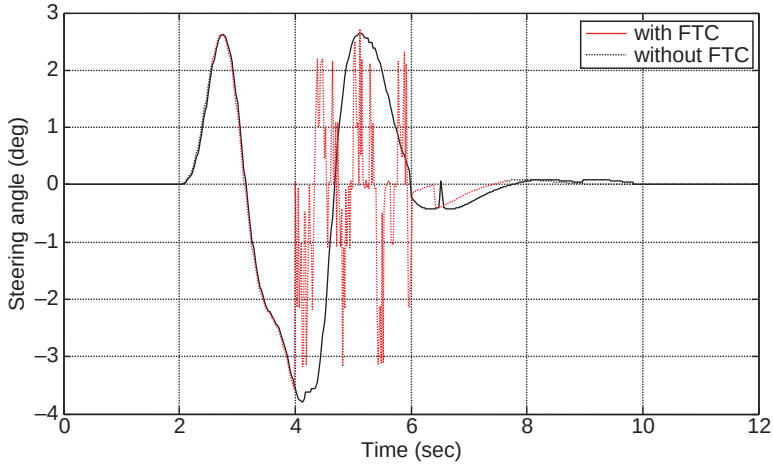

Figure 4. Steering angle with 90% loss at $v = 30$ m/s and $u = 1$.

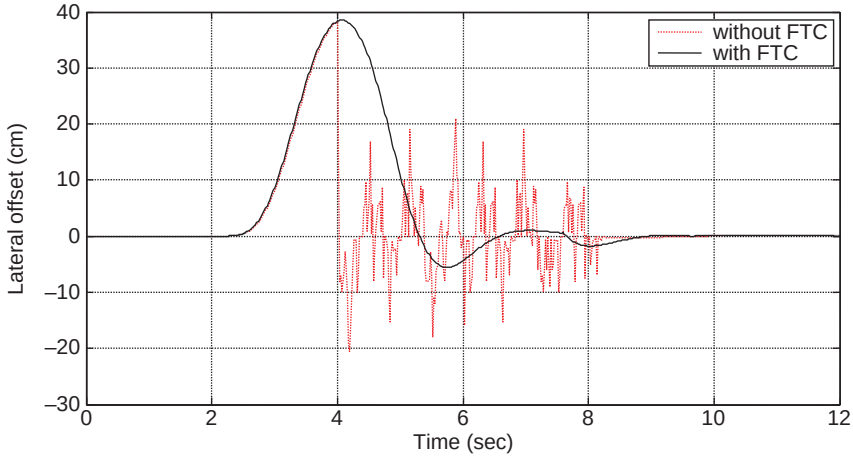

Figure 5. Steering angle with 90% loss at $v = 40$ m/s and $u = 0.75$.

control loop. Therefore, quick accommodation of faults in the lane-keeping control system is an important issue. Note though that due to the increase in speed and decrease in the road adhesion factor, the vehicle takes more time to follow the path. In contrast, the vehicle is not able to track the set path after the fault occurrence in the case without FTC.

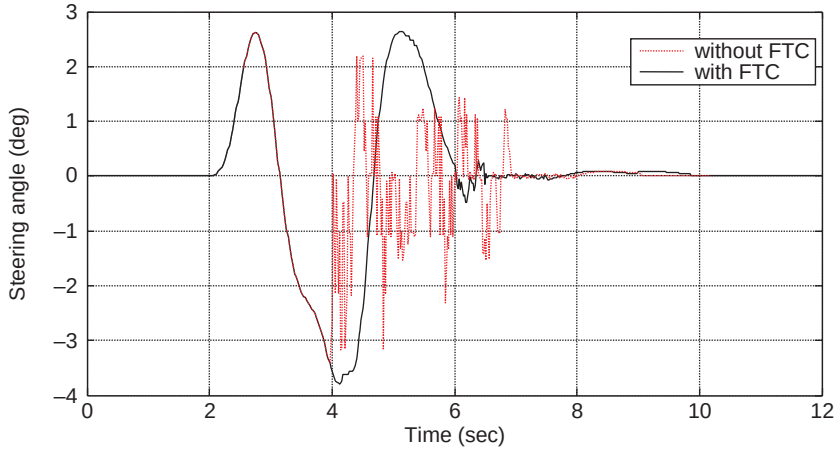

Figure 6. Steering angle with 90% loss at $v = 40$ m/s and $u = 0.75$.

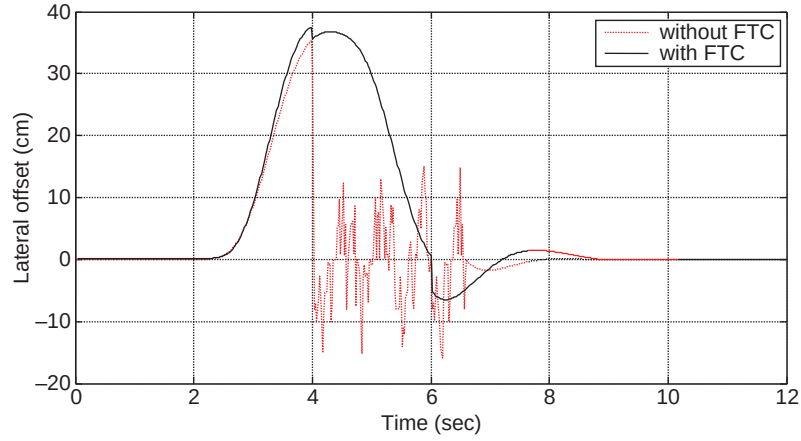

Figure 7. Lateral offset with 70% sensor information loss at $v = 40$ m/s and $u = 0.75$.

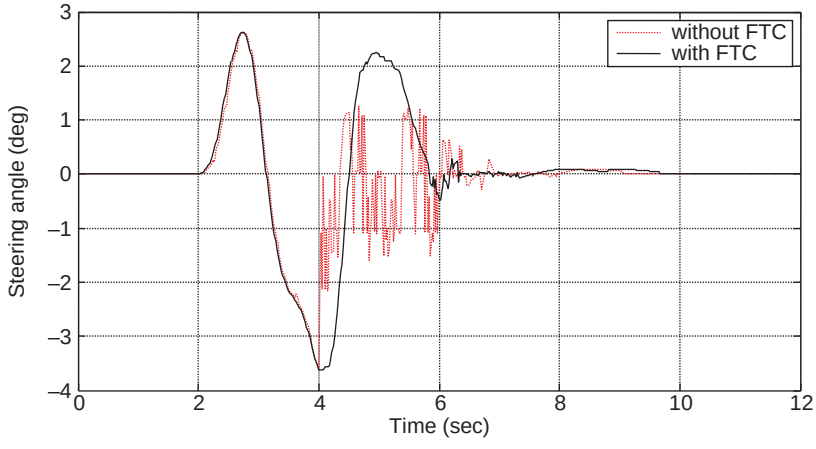

Figure 8. Steering angle at 70% sensor information loss at $v = 40$ m/s and $u = 0.75$.

Next, a 70% sensor information loss is considered ($a = 1$). The observer gains in this case were $L_1 = 8$ and $L_2 = 10$, while the controller gain was $K = 50$. Figures 7 and 8 illustrate the response of the vehicle with a longitudinal speed of 40 m/s (108 km/h) and considering a road adhesion factor of one.

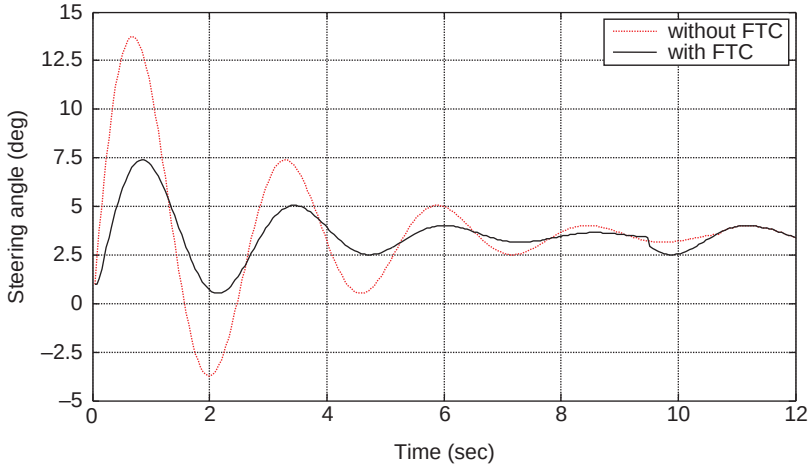

Figure 9. Lateral offset on a circular track with $v = 30$ m/s and $u = 1$.

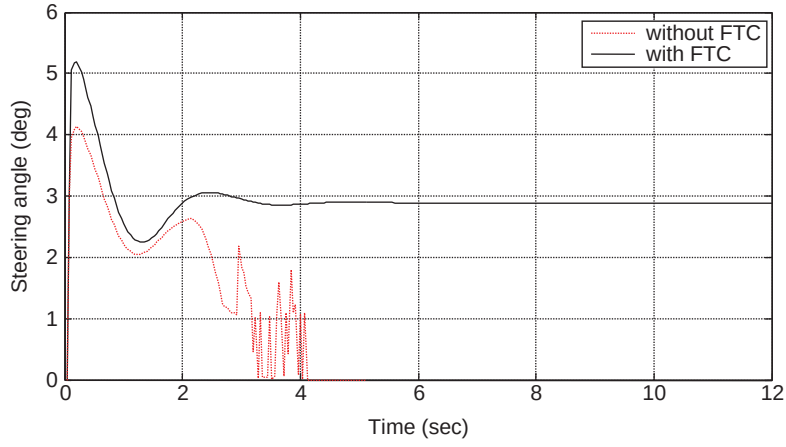

Figure 10. Steering angle on a circular track with $v = 30$ m/s and $u = 1$.

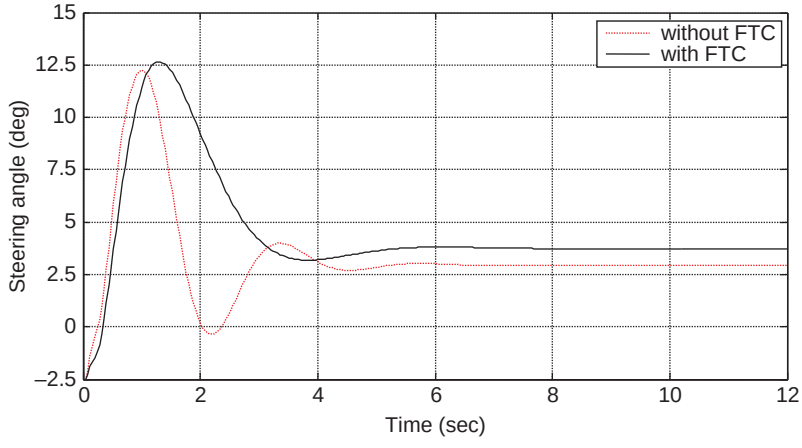

Figure 11. Lateral offset at 70% sensor information loss with $v = 30$ m/s and $u = 1$.

In this case, as expected fewer disturbances can be observed in the presence of faults when compared to the case with 90% information loss.

4.2. Circular track

In this section, we consider a circular track of 500 ft in order to show the performance of the observer-based FTC

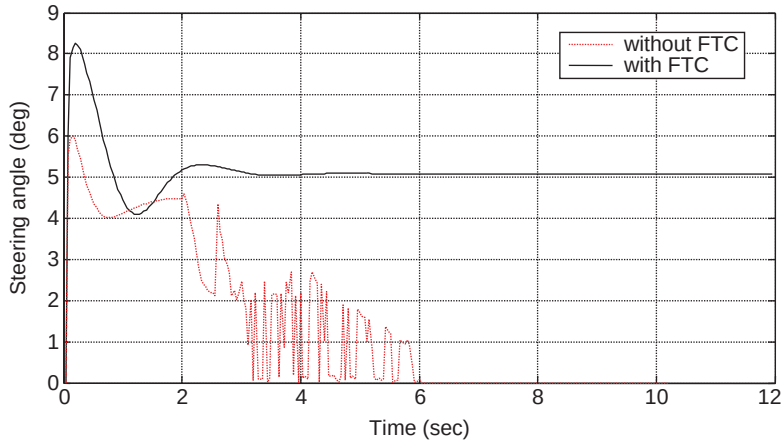

Figure 12. Steering angle at 70% sensor information loss with $v = 30$ m/s and $u = 1$.

algorithm. This path can provide valuable insight into the handling of a vehicle as well as some important characteristics of the proposed controller. Hence, this path was chosen to illustrate the steady-state characteristics of the controller while executing a constant nonzero curvature path. Experiments on this path are performed at a constant longitudinal speed of 30 m/s (108 km/h) and considering a road adhesion factor of one. Figures 9 and 10 illustrate the response of the vehicle when considering 90% of sensor information loss.

Note that information from the sensor is lost after 2 s and the vehicle requires more time to follow the path, as shown in Figure 9. The steering controller is stabilized after 4 s and system stability is maintained as shown in Figure 10. Next, the simulation was carried out with 70% sensor information loss. Figures 11 and 12 illustrate the lateral offset and steering angle in such case.

The simulation results show the effectiveness of the proposed FTC algorithm in improving the vehicle response under different paths and using various driving scenarios under several faulty conditions. Note that compared to the case without an FTC algorithm, the system is not able to recover from the sensor fault and the controller loses its steering capabilities shortly after the occurrence of the sensor fault. This is more prominent when the vehicle is following a circular path.

5. Conclusion

This paper presented an effective FTC paradigm that integrates the optimal properties of the LQR framework with an observer-based FD scheme to maintain vehicle stability and ensure handling in the presence of faults. A weight adjustment algorithm is incorporated in the FDI unit to ensure robust performance in the presence of parameter variations and disturbances. For accurate validation, the control routines were implemented in MATLAB/Simulink environment and tested using CarSim, a high-fidelity vehicle simulator. The results confirmed the ability of the proposed FTC framework to effectively monitor the system and ensure correct tracking performance under faulty conditions. Future work will focus on the integration of the proposed methodology with environment information devices such as radars and vision systems.

Funding

This work is partially supported by the Louisiana Board of Regents Support Fund contract numbers LEQSF (2012-15)-RD-A-26 and LEQSF-EPS (2015)-PFUND-421 and by LaSPACE/NASA [grant number NNX10AI40H] sub awards No. 84415 and No. 89632.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Arogeti, S., Wang, D., Low, C., & Yu, M. (2008, January). Fault detection isolation and estimation in a vehicle steering system. *IEEE Transaction in Industrial Electronics*, *59*(12), 1–2.

Blanke, M., & Staroswiecki, M. (2006). *Diagnosis and fault-tolerant control* (2nd ed.). Berlin: Springer-Verlag.

Chen, H., Song, Y., & Li, D. (2011). *Fault-tolerant tracking control of FW-steering autonomous vehicles*. Proceedings of the Chinese Control and Decision Conference CCDC, Mianyang, June 25–27, 2008, pp. 92–97.

Cheng, H. (2011). *Autonomous intelligent vehicles. Theory, algorithms and implementation*. London: Springer-Verlag.

Dong, J., Verhaegen, M., & Holweg, E. (2008, July 6–11). *Closed-loop subspace predictive control for fault tolerant MPS design*. Proceedings of the 17th IFAC World Congress, Seoul, pp. 3216–3221.

Fekih, A. (2014, June). *Fault diagnosis and fault tolerant control design for aerospace systems: A bibliographical review*. Proceedings of the 2014 American Control Conference, Portland, OR, pp. 1286–1291.

Fekih, A., & Deveriste, D. (2013, June 17–19). *A Fault-Tolerant Steering Control Design for Automatic Path Tracking in Autonomous Vehicles*. Proceedings of the American Control Conference, Washington, DC.

Fekih, A., & Seelem, S. (2012, October 3–5). *A fault tolerant control design for automatic steering control of ground vehicles*. Proceedings of IEEE Multi-Conference on Systems and Controls, Croatia, Dubrovnic, pp. 1491–1496.

Gustaffson, F. (2000). *Adaptive filtering and change detection*. Chichester: John Wiley & Sons.

Herpin, J., Fekih, A., Golconda, S., & Lakhotia, A. (2007, December). Steering control of the autonomous vehicle: CajunBot. *AIAA Journal of Aerospace Computing, Information, and Communication*, *4*, 1134–1142.

Hsiao, T., & Tomizuka, M. (2005, June 8–10). *Threshold selection for timely fault detection in feedback control systems*. Proc. of the American Control Conference, Vol. 5, Portland, pp. 3303–3308.

Jiang, J., & Yu, X. (2012, April). Fault tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in Control*, *36*(1), 60–72.

Laureiro, R., Benmoussa, S., Touati, Y., Merzouki, R., & Ould-Bouamama, B. (2014, January). Integration of fault diagnosis and fault-tolerant control for health monitoring of a class of MIMO intelligent autonomous vehicles. *IEEE Transaction on Vehicular Technology*, *63*(1), 30–39.

Mechanical Corporation. Retrieved from www.carsim.com

Morteza, M., & Fekih, A. (2014a). A stability guaranteed robust fault tolerant control design for vehicle suspension systems subject to actuator faults and disturbances. *IEEE Transactions on Control Systems Technology*, *PP*(99), 1–10.

Morteza, M., & Fekih, A. (2014b, March). Adaptive PID-sliding mode fault tolerant control approach for vehicle suspension systems subject to actuator faults. *IEEE Transactions on Vehicular Technology*, *63*(3), 1041–1054.

Noura, H., Theilliol, D., Ponsart, J.-C., & Noura, A. (2009). *Fault-tolerant control systems. Design and practical applications*. London: Springer-Verlag.

Tabbache, B., Benbouzid, M., Kheloui, A., & Bourgeot, J. (2011, June 27–30). *DSP-based sensor fault-tolerant control of electric vehicle power trains*. Proceedings of IEEE International Symposium on Industrial Electronics, Gdansk, pp. 2085–2090.

Wang, R., & Wang, J. (2013, March). Passive actuator fault-tolerant control for a class of over-actuated nonlinear systems and applications to electric vehicles. *IEEE Transaction on Vehicular Technology*, *62*(3), 972–985.

Yang, H., Cocquempot, V., & Jiang, B. (2008, June 25–27). *Hybrid fault tolerant tracking control design for electric vehicles*. Proceedings of the Mediterranean Conference on Control and Automation, Ajaccio, pp. 1210–1215.

Zhang, M., Ding, S., Lam, J., & Wang, H. (2003, March). An LMI approach to design robust fault detection filter for uncertain LTI systems. *Automatica*, *39*(3), 543–550.

Appendix

A.1. CarSim steering controller

The theory and the algorithm used to steer a road vehicle in CarSim environment are described in this section. The algorithm is intended to provide optimal control for a continuous linear system:

$$\dot{x} = Ax + Bu + Hv \tag{A1}$$

$$y = Cx + Du + Ev. \tag{A2}$$

where x is an array of n state variable, u is control input, y is an output, and A, B, C, D, E, and H are matrices with constant coefficients. The control objective is to determine the value of u to predict output to $y(t)$ to match a target $y_{\text{target}}(t)$ over some preview time t.

The system has initial conditions x_0 at time $t = 0$, a constant input u, and a constant disturbance v, then the time response is defined by

$$x(t) = e^{At}x_0 + \int_0^t e^{A\eta}Bud\eta + \int_0^t e^{A\eta}Hvd\eta. \tag{A3}$$

The term $e^{A\eta}$ is an $(n \times n)$ matrix called the state transition matrix. Each coefficient in the matrix is the portion of state variable i at time t that is linearly related to the state variable j at time zero. The two integrals in Equation (A3) define the forced responses to each state variable due to constant control u and disturbance v over the time interval.

Combining Equations (A2) and (A3), we obtain the following output response:

$$y(t) = Cx = Ce^{At}x_0 + C\left[\int_0^t e^{A\eta}\mathrm{d}\eta\right][Bu + Hv]. \tag{A4}$$

A control response scalar $g(t)$ and a disturbance response scalar $h(t)$ are defined to relate the responses over the time t. A free response array F is defined and relates the state variables at time 0 to the resulting output variable y at time t.

$$F(t) = \mathrm{Ce}^{\mathrm{At}} \tag{A5}$$

The response equation is defined by

$$y(t) = F(t)x_0 + g(t)u + h(t)v \tag{A6}$$

To determine the optimal control, a quadratic performance index J is defined by

$$J = \frac{1}{T}\int_0^T \{y_{\text{target}}(t) - y(t)\}^2 W(t)\mathrm{d}t. \tag{A7}$$

Here, $W(t)$ is an arbitrary weighting function. A optimal control law is designed by minimizing the cost function J, representing the squared deviation of response variable $y(t)$ relative to the target function $y_{\text{target}}(t)$. The control function u minimizing J can be found by substituting Equation (A6) into (A7) and taking a partial derivative of J with respect to u.

$$J = \frac{1}{T}\int_0^T (F(t)x_0 + g(t)u + h(t)v - y_{\text{target}}(t))^2 W(t)\mathrm{d}t. \tag{A8}$$

Solving for u results in

$$u = \frac{\int_0^T \{y_{\text{target}}(t) - F(t)x_0 - h(t)v\}g(t)W(t)\mathrm{d}t}{\int_0^T g(t)^2 W(t)\mathrm{d}t}. \tag{A9}$$

In practice, the integrals over T can be replaced with finite summations

$$u = \frac{\sum_{i=1}^m (y_{\text{target}}(t) - F_i x_0 - h_i v)g_i W_i}{\sum_{i=1}^m g_i^2 W_i}. \tag{A10}$$

The algorithm is programmed to generate a steering wheel angle in the vehicle solver program for a given target path. The algorithm synthesizes the target path over the preview time and calculates the optimal front steering effort u to minimize deviations from the path. It also delays the driver steering control by a constant time. The geometry of the road is given as a sequence of X and Y coordinates that define a reference line. Station S is defined as the distance along the reference line, typically a road

centerline. For each pair of X–Y coordinates, a corresponding increment of S is computed by

$$S_i = S_{i-1} + \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2}. \quad \text{(A11)}$$

To calculate the optimal steering control using Equation (A10), the target position is needed at each point considered in the summation. The station target location is

$$S_{\text{targ},i} = S + \frac{iV_xT}{m}, \quad \text{(A12)}$$

where V_x is the vehicle forward speed.

The controller calculations are performed using an axis system where the vehicle is located such that the center of the vehicle front axle is at $X = 0$ and $Y = 0$ and the X and Y axes are aligned with the longitudinal and lateral axes of the vehicle. The target lateral translation is calculated by first getting the inertial X and Y coordinates of the path as a function of the station at the target location (S_{targ}):

$$Y_{\text{target}} = [Y(S_{\text{targ}}) - Y_V]\cos(\psi) - [X(S_{\text{targ}}) - X_V]\sin(\psi). \quad \text{(A13)}$$

A.2. Vehicle parameters

Table A1. Vehicle parameters.

M	Vehicle mass, 1573 kg
c_f, c_r	Cornering stiffness of front/rear wheels 2*60,000 N/rad
l_f	Distance between the front wheels and the center of gravity, 1.137 m
l_r	Distance between the rear wheels and the center of gravity, 1.530 m
I_z	Yaw moment of inertia, 2753 kg m^2