Constraint Management in Fuel Cells: A Fast Reference Governor Approach

Ardalan Vahidi   Ilya Kolmanovsky   Anna Stefanopoulou

Abstract— The air supply system in a fuel cell may be susceptible to saturation during transient driving conditions. The air compressor surge and choke can disrupt the flow of air into the cathode and negatively impact fuel cell power generation. Low partial oxygen pressure in the cathode can damage the stack and reduce its life. A load governor, added to the air supply control system, can monitor the transients and prevent constraint violation by modifying the reference current command to the fuel cell stack. In this paper we develop such a load governor using two approaches. The first approach is based on on-line model predictive control and the second approach utilizes a fast reference governor. We discuss the performance and computational requirements of each method. We propose a modification to the fast reference governor design to make it applicable to a nonlinear plant.

I. INTRODUCTION

In fuel cell powered vehicles, one of performance bottlenecks is posed by the air supply system. In a high pressure Proton Exchange Membrane (PEM) fuel cell, a compressor supplies air to the cathode. The compressor itself consumes up to 30% of fuel cell generated power and therefore its size has direct influence on overall system efficiency. More importantly the compressor performs the critical task of providing the oxidant into the stack. It is known in the fuel cell community that low partial oxygen pressure in the cathode reduces the fuel cell voltage and the generated power, and it can reduce the life of the stack [1]. The challenge is that oxygen reacts instantaneously as current is drawn from the stack, while the air supply rate is limited by the manifold dynamics and compressor surge and choke constraints [2], [3]. Surge causes large variations in flow and sometimes flow reversal through the compressor. Large amplitude surge may even damage the compressor. Reference [4] develops an active surge control approach for centrifugal compressors. Choke happens at sonic mass flow and is an upper limit to the amount of air the compressor can provide. In the fuel cell system there is a potential for compressor choke during a step-up in current demand, and there is a potential for compressor surge during a step-down in current demand. For instance, the air flow controller reduces the compressor motor voltage during a step-down in current demand. A sudden decrease in compressor motor voltage is followed by a fast decrease in the compressor rotational speed. Since the manifold pressure cannot drop as quickly, surge may occur.

Low-pass filtering of the current demand to a fuel cell during step-downs in current demands can prevent surge. The design of these filters is usually conservative to ensure satisfactory operation under various operating conditions (see e.g., [5]). The transients

A. Vahidi (corresponding author) is with the University of Michigan, G008 Walter E. Lay Auto Lab, 1231 Beal Avenue, Ann Arbor, MI, 48109, E-mail: `avahidi@umich.edu`.

I.V. Kolmanovsky is with Ford Research and Advanced Engineering, Ford Motor Company, Dearborn, MI 48124, E-mail: `ikolmano@ford.com`.

A. Stefanopoulou is with the University of Michigan, G058 Walter E. Lay Auto Lab, 1231 Beal Avenue, Ann Arbor, MI, 48109, E-mail: `annastef@umich.edu`.

may be managed with a load governor which modifies the current drawn from the fuel cell by only as much as needed for constraint enforcement.

Sun and Kolmanovsky [5] have developed a load governor for starvation prevention in a fuel cell using a nonlinear reference governor approach. Their reference governor searches at each sample time instant for the optimal and constraint admissible current demand to the fuel cell based on on-line optimization of a scalar parameter and on-line simulations of the nonlinear fuel cell model; their approach ensures robustness against parameter variations. To implement the load governor in memory- and chronometric-constrained automotive microcontrollers, it is desirable to reduce the online computational effort as well as RAM (random-access memory) and ROM (read-only memory) requirements.

In [6] a methodology for fast reference governor (FRG) design for linear systems with pointwise-in-time state and control constraints was developed. It was shown that the reference governor can be constructed largely off-line, while the on-line computational effort can be systematically reduced at the expense of the increased conservatism in the reference governor operation.

In the present paper we develop an FRG-based load governor for the fuel cell system and we illustrate its performance and computational requirements. For comparison, we also develop a model predictive control (MPC) based load governor. Finally, we discuss modifications to the FRG-based load governor design so that it can be applied to the nonlinear fuel cell model.

We next briefly describe the fuel cell model.

II. MODEL OF THE FUEL CELL SYSTEM

A nonlinear spatially-averaged model of a 75kW fuel cell stack together with its auxiliaries is developed in [7] based on electrochemical, thermodynamic and fluid flow principles. The fuel cell has 350 cells and can provide up to 300 A of current. Its model, representing membrane hydration, anode and cathode flow and stack voltage, is augmented with the models of ancillary subsystems including the compressor, manifold dynamics, cooling system and the humidifier to obtain a nonlinear model of the overall fuel cell system. Since the focus of this paper is on control of air flow, we present the governing equations, essential to understanding the dynamics between the compressor and the air flow into the cathode. The interested reader is referred to [7] for additional details.

Low partial oxygen pressure in the cathode decreases the fuel cell voltage and the generated power and can thus reduce the life of the stack. To prevent such a situation the oxygen level in the cathode needs to be regulated. The oxygen excess ratio (OER), is defined for this purpose as follows [7]:

$$\lambda_{O_2} = \frac{W_{O_2,in}}{W_{O_2,rct}}, \tag{1}$$

where $W_{O_2,in}$ is the flow of oxygen into the cathode and $W_{O_2,rct}$ is the mass of oxygen reacted in the cathode. Low values of λ_{O_2} indicate oxygen starvation. The rate of oxygen reacted $W_{O_2,rct}$, depends on the current drawn from the stack I_{fc}:

$$W_{O_2,rct} = M_{O_2} \frac{nI_{fc}}{4F}, \tag{2}$$

where n is the number of cells in the stack, F is the Faraday number, and M_{O_2} is the oxygen molar mass. Therefore increase in current drawn from the fuel cell, λ_{O_2} is decreased. To maintain the level of oxygen excess ratio, more air should be supplied to the fuel cell. The flow rate of the oxygen into the stack $W_{O_2,in}$, is a function of the air flow out of the supply manifold W_{sm}:

$$W_{O_2,in} = y_{O_2} \frac{1}{1+\Omega_{atm}} W_{sm}, \tag{3}$$

where $y_{O_2} = 0.21 \frac{M_{O_2}}{M_a^{atm}}$ is the mass ratio of oxygen in the dry atmospheric air and Ω_{atm} is the humidity ratio of the atmospheric air. The mass flow rate out of the supply manifold W_{sm}, depends on the downstream (cathode) pressure and upstream (supply manifold) pressure p_{sm}, and temperature T_{sm}, through an orifice equation. The cathode total pressure depends on the partial pressure of the (i) oxygen which is supplied $W_{O_2,in}$, oxygen which is reacted $W_{O_2,rct}$, and the oxygen removed, (ii) nitrogen which is supplied and removed and (iii) the water which is supplied, generated, transported through the membrane and removed. The additional cathode states of oxygen mass m_{O_2}, nitrogen mass m_{N_2}, water vapor mass $m_{w,ca}$, total return manifold pressure p_{rm}, and anode states of hydrogen mass m_{H_2}, and water vapor $m_{w,an}$, are needed to capture the temporal dynamics of the total cathode pressure during a step change in current. These detailed state equations are omitted here but can be found in [7]. However, to allow the reader understand how the control input affects the supply manifold flow W_{sm}, we add the following relations. Specifically the supply manifold pressure p_{sm}, and mass m_{sm}, are related to the compressor's air flow W_{cp}, and temperature T_{cp}, through the following dynamics:

$$\frac{dp_{sm}}{dt} = K_{sm}(W_{cp}T_{cp} - W_{sm}T_{sm}), \tag{4}$$

$$\frac{dm_{sm}}{dt} = W_{cp} - W_{sm}, \tag{5}$$

where K_{sm} is a coefficient determined by air specific heat coefficients and the manifold volume. The vehicle configuration captured in our model is the Ford P2000 [8] which has a large supply manifold due to location of the air compressor relative to the fuel cell. The supply manifold temperature T_{sm} is defined by the ideal gas law. The compressor air flow W_{cp} and its temperature T_{cp} depend on the compressor rotational speed ω_{cp}:

$$J_{cp}\frac{d\omega_{cp}}{dt} = \frac{1}{\omega_{cp}}(P_{cm} - P_{cp}), \tag{6}$$

where J_{cp} is the compressor inertia and P_{cp} is the power absorbed by the compressor. The power supplied to the compressor P_{cm}, is a function of compressor motor voltage V_{cm}.

In summary, the compressor voltage V_{cm}, controls the speed of the compressor through the first-order nonlinear dynamics shown in (6). The speed of the compressor determines the compressor flow rate W_{cp}, which then according to equation (4), affects the supply manifold pressure p_{sm}. The latter, together with the cathode pressure, determines the supply manifold flow W_{sm}, and finally the flow rate of the oxygen into the cathode $W_{O_2,in}$.

To summarize, the fuel cell system model is described by a set of first-order nonlinear differential equations:

$$\begin{aligned}
\dot{x}_{nl} &= h(x_{nl}, u, v), \\
u &= [V_{cm}], \\
v &= [I_{fc}], \\
y &= [W_{cp} \quad p_{sm} \quad \lambda_{O_2}],
\end{aligned} \tag{7}$$

where x_{nl} is the state vector of the nonlinear dynamic system. The compressor motor command V_{cm}, is the control input, and the current drawn from the fuel cell, I_{fc}, is a measured disturbance.
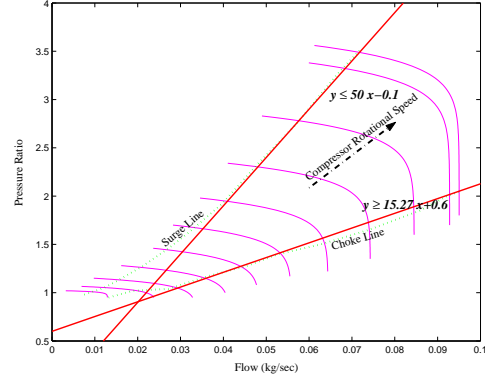

Fig. 1. A compressor map with approximate surge and choke boundaries.

The outputs of interest are air flow rate through the compressor W_{cp}, supply manifold pressure p_{sm}, and oxygen excess ratio λ_{O_2}[1]. For the control design purpose, this augmented nonlinear system is linearized at a representative operating point. We choose nominal stack current as $I_{fc}^0 = 192$ A. The nominal value for oxygen excess ratio is selected at $\lambda_{O_2}^0 = 2.0$, which corresponds to maximum fuel cell net power for the nominal current [7]. The compressor motor voltage needed, to supply the optimum air flow that corresponds to I_{fc}^0 and $\lambda_{O_2}^0 = 2.0$, is $V_{cm}^0 = 164$ volts. The linearized system has eight dynamic states and is described by:

$$\begin{aligned}
\dot{x}_{ln} &= Fx_{ln} + G_u u + G_v v, \\
y &= Cx_{ln} + D_u u + D_v v,
\end{aligned} \tag{8}$$

where the variables x_{ln} and y show deviations from their nominal values. The linear state vector is:

$$x_{ln} = \delta[m_{O_2} \quad m_{H_2} \quad m_{N_2} \quad w_{cm} \quad p_{sm} \quad m_{sm} \quad m_{w,an} \quad p_{rm}]^T,$$

where δ stands for the deviation from the operating point. A discrete-time version of this linear model is used in the paper for load governor design. The nonlinear model (7) is used in nonlinear closed-loop simulations.

Figure 1 shows a compressor map with superimposed surge and choke constraint lines. In this map each solid line curve represents a compressor rotational speed. The surge and choke boundaries are represented by dotted lines. We want to introduce constraints which ensure operation of the compressor between the surge and choke lines and prevent stack starvation. These constraints can be best described as output constraints[2]. The nonlinear surge boundary can be approximated by a straight line for most part of the operating region as shown in Fig. 1. Both compressor flow and pressure ratio are functions of states of the system and are relatively easy to measure. Choke limit can be expressed similarly. The constraints can then be represented by two linear inequalities defined by the linear approximation:

$$\begin{aligned}
-0.0506\delta W_{cp} + \delta p_{sm} &\leq 0.4, \\
0.0155\delta W_{cp} - \delta p_{sm} &\leq 0.73.
\end{aligned} \tag{9}$$

[1]Oxygen excess ratio cannot be directly measured in practice. Instead the stack voltage V_{st}, can be measured. An observer can be designed to reconstruct oxygen excess ratio from measurements of air flow rate through the compressor W_{cp}, supply manifold pressure p_{sm}, and stack voltage V_{st} as shown in [7].

[2]An initial attempt to enforce slew rate constraint on the compressor input for preventing surge reduced the excursion into the surge region, but surge still occurred for relatively large transients. Moreover the slew rate limit resulted in oscillations in control command and poor closed-loop performance.
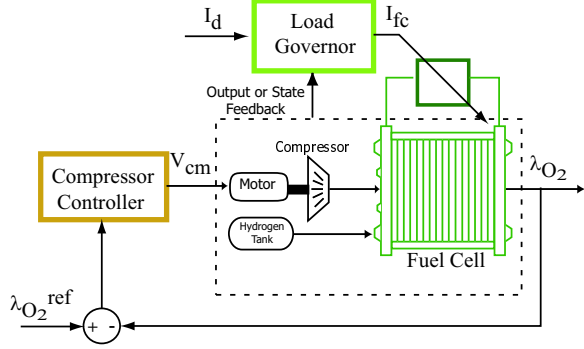
Fig. 2. Schematic of Fuel Cell Air Supply Control.

We limit the lower limit of oxygen excess ratio to above 1.9 to avoid stack starvation:

$$\lambda_{O_2} \geq 1.9 \Rightarrow \delta\lambda_{O_2} \geq -0.1. \tag{10}$$

We now proceed to design a load governor which enforces these constraints through the modification of the current demand.

III. REFERENCE MODIFICATION FOR CONSTRAINT SATISFACTION

A schematic of fuel cell air supply control system is shown in Figure 2. The compressor controller is a linear controller and regulates the supply of oxygen to the cathode during changes in current demand[3]. Proper design renders a stable closed-loop system, in the vicinity of an operating equilibrium. Still, a rapid transient in current can send the compressor into surge or choke and saturate the air supply system or result in oxygen starvation in the cathode. Extra measures need to be taken to ensure desired operation of the system. A load governor, shown in Fig. 2, is an add-on device that slows the transitions in current demand I_d, so that the constraints are not violated. A load governor, could be as simple as a first order filter which is designed for the worst-case current demand. The drawback of such a "passive" filter is that it slows down the system response even under small transients. The transients can be managed less conservatively with an "active" load governor which modifies the reference command only when needed to avoid constraint violation. The problem can be posed as an optimization problem, one in which the difference between the demand and fuel cell current is minimized over a future horizon, subject to pointwise-in-time constraints.

A discrete-time model of the closed-loop system is used for predicting the plant response:

$$\begin{aligned} x(k+1) &= A_{cl}x(k) + B_v I_{fc}(k), \\ y(k) &= Cx(k) + D_v I_{fc}(k). \end{aligned} \tag{11}$$

The governed current I_{fc}, is the optimization variable which needs to be determined. The assumption is that measured current demand $I_d(k)$, stays constant over the future horizon. The problem can then be formulated using two frameworks:

> I) Model Predictive Control (MPC): At each time instant k, a sequence $\begin{bmatrix} I_{fc}(k) & I_{fc}(k+1) & \dots & I_{fc}(k+N) \end{bmatrix}^T$ is sought which satisfies the constraints and minimizes a quadratic cost function;
> II) I_{fc} is assumed to remain constant over the future horizon, $I_{fc}(k) = I_{fc}(k+1) = \dots = I_{fc}(k+N)$.

As shown in [6], the second approach reduces to a scalar optimization at each time instant. A solution to such a problem is developed in [6], where a large portion of computation is carried off-line

---

[3]To prevent pressure difference across the membrane, a solenoid valve regulates hydrogen pressure in the anode.

rendering the online part less demanding. Next we elaborate on each approach and evaluate the performance and computational requirements of each scheme.

A. MPC-Based Load Governor

In the MPC approach, a performance index is formulated to minimize the difference between the current demand and fuel cell current, I_d and I_{fc} respectively, i.e.,

$$J(k) = \sum_{j=1}^{p} \|(I_d(k) - I_{fc}(k+j))\|^2, \tag{12}$$

subject to pointwise-in-time constraints. In the performance index, p is the prediction horizon. We seek a sequence $[I_{fc}]_k^* = \begin{bmatrix} I_{fc}^*(k) & I_{fc}^*(k+1) & \dots & I_{fc}^*(k+p) \end{bmatrix}^T$ which minimizes the performance index subject to three linear constraints given in equations (9) and (10) applied along the prediction horizon. In summary:

$$\begin{bmatrix} -0.0506 & 1 & 0 \\ 0.0155 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} y(k+j|k) \leq \begin{bmatrix} 0.4 \\ 0.73 \\ 0.1 \end{bmatrix}, \quad j = 1,2,\dots,p, \tag{13}$$

where $y(k+j|k)$ is the predicted value of the outputs at instant $k+j$ based on information available at instant k. The performance index (12) and the constraints (13) can be rearranged into a quadratic function of the variables I_{fc}^*, and initial conditions. Quadratic programming (QP) techniques can be used to solve this constrained optimization problem at each sampling time instant. The number of free variables to be determined is equal to the number of prediction steps p, while the total number of constraints is $3p$.

B. FRG-Based Load Governor

In the FRG approach, the dimensionality of the optimization problem is reduced with the assumption that I_{fc} stays constant over the prediction horizon. The reference modification can be accomplished via a first order linear filter with a scalar adjustable bandwidth parameter, β:

$$I_{fc}(k+1) = I_{fc}(k) + \beta(k)(I_d(k) - I_{fc}(k)), \tag{14}$$

and $\beta(k) \in [0,1]$. Ideally $\beta(k) = 1$ meaning $I_{fc}(k+1) = I_d(k)$ and the current command only suffers a unit delay. When there is a large change in I_d and a possibility of future constraint violation exists, β is reduced to avoid constraint violation. In the extreme case when $\beta(k) = 0$, we have $I_{fc}(k+1) = I_{fc}(k)$. The parameter $\beta(k)$ is maximized at each sample time k, subject to the condition that maintaining $I_{fc}(j) = I_{fc}(k)$ for all $j > k$ guarantees constraint satisfaction.

The optimization can be solved in a few different ways. It can be arranged as a linear programming (LP) problem with the single variable β and the constraints (13) for a sufficiently large p, and solved online. Bisectional search for maximum constraint-admissible β is another possible online solution and is applicable to nonlinear systems as well [5]. Such online solutions may be computationally intensive for systems with more than a few states.

Fortunately for linear systems, a large portion of calculations can be performed off-line, thereby reducing the online computational effort. Specific procedures for such fast reference governors are detailed in [6] and they are used here for the fuel cell application. We provide a brief summary of the methodology and refer the interested reader to [6] for details.

Future constraint violations can be predicted by checking if the state of the system belongs to a maximal output admissible set, called O_∞. The O_∞ is the set of all initial states $x(0)$, and the

modified reference $I_{fc}(0)$, which with $\beta = 0$ guarantee satisfaction of constraints in future. It is defined as:

$$O_\infty = \{(x(0), I_{fc}(0)) : y(k) = Cx(k) + D_v I_{fc} \in Y \quad \forall k \in Z^+\}, \quad (15)$$

where Y is the constraint set described by equation (13) and the state dynamics are those of (11). The set, O_∞, is positively invariant for the system defined by (11) with $\beta = 0$. Thus if the system starts in this set and the current I_{fc} is kept constant into the future, the trajectory will remain in O_∞ and the constraints will be satisfied.

The goal is to find the maximum value of β which maintains the state in O_∞:

$$\beta^* = max\{\beta \in [0,1] : \begin{bmatrix} Ax + B_v I_{fc} \\ I_{fc} \end{bmatrix} + \beta \begin{bmatrix} 0 \\ I_d - I_{fc} \end{bmatrix} \in O_\infty\}. \quad (16)$$

The set O_∞ does not, in general, admit a characterization by a finite set of linear inequalities (i.e., it may not be finitely determined). It does, however, have a computable approximation, $\tilde{O}_\infty$, which is finitely determined, see [6]. For a system with n states and linear constraints, $\tilde{O}_\infty \subset R^{n+1}$ is a polytope with m faces represented as a set of solutions to a system of linear inequalities of the form:

$$\Phi_{m \times n+1} \begin{bmatrix} x(0) \\ I_{fc}(0) \end{bmatrix} \leq \Theta_{m \times 1}. \quad (17)$$

For systems with large state dimension and high sampling rates the number of inequalities in the representation of $\tilde{O}_\infty$ can grow large. This is undesirable for two reasons: The effort to compute $\beta(k)$ increases with the number of inequalities in the representation of $\tilde{O}_\infty$ and ROM size to store a representation of $\tilde{O}_\infty$ also increases with the number of inequalities in its representation. Often, however, some inequalities in the representation of $\tilde{O}_\infty$ are almost redundant, i.e., if they are eliminated from the representation of $\tilde{O}_\infty$ the resulting polytope is only slightly larger than $\tilde{O}_\infty$. Since the polytope resulting from such constraint elimination may not be a constraint-admissible set of initial conditions itself, it is scaled down uniformly in the x-direction (but not in the I_{fc} direction) until it is contained in $\tilde{O}_\infty$. After this process of inequality elimination and shrinkage, we obtain a polytope P, which is constraint admissible and has fewer inequalities compared to $\tilde{O}_\infty$; at the same time, one has to keep in mind that P is only a subset of $\tilde{O}_\infty$ and thus can result in more conservative performance.

The set P may not be positively invariant. Thus the situation may arise that a feasible $\beta \in [0,1]$ does not exist. In this case, in agreement with the theoretical results in [6], the reference governor sets $\beta = 0$.

Once $\tilde{O}_\infty$ or P are determined, the online evaluation of β^* is relatively simple. A computationally efficient method for finding β^* is given in [6] which involves a fixed number of adds, subtracts, divides, multiples, max and min operations in each sampling interval.

We next evaluate the performance of the FRG-based load governor and we compare its performance and computational requirements with the MPC-based load governor.

IV. SIMULATION ANALYSIS

In this section we analyze the performance of the FRG-based load governor and compare it to the MPC-based load governor. The sampling frequency is fixed at 100 Hz. Constraints are surge, choke and starvation constraints given in (13). For MPC-based design the prediction horizon is 10 sampling intervals. At each step a quadratic program with 10 variables and $3 \times 10 = 30$ constraints is solved. For the fast reference governor, $\tilde{O}_\infty \subset R^{11}$ is characterized by 348 linear inequalities ($m = 348$) and is determined offline in 140 steps. Constraint elimination and a shrinkage factor of 1.2 generate $P \subset \tilde{O}_\infty$ with only 77 linear inequalities. For comparison, shrinkage factors of 1.4, 1.75, 2.0, 10 resulted in P with,
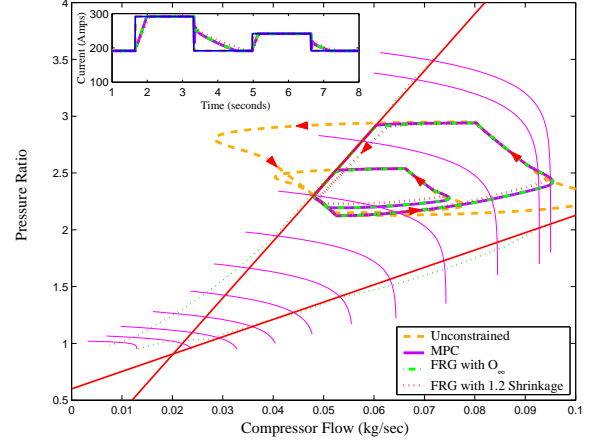

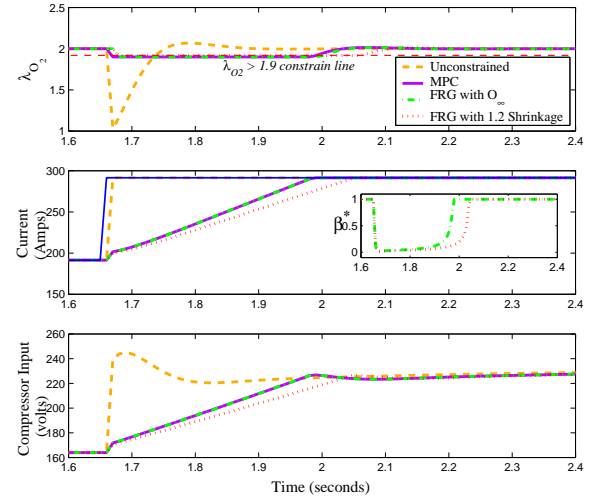Fig. 3. Compressor flow trajectory for different load governor designs.

Fig. 4. Simulation during tip-in transient when oxygen starvation constraint is active.

respectively, 65, 51, 42, and 22 inequalities; the P corresponding to 1.2 was selected as the best compromise between the number of inequalities and conservatism of the reference governor operation.

We examine the performance of different governor configurations during a series of step-ups and step-downs in the current demand (which correspond to driver tip-in and tip-out commands, respectively) with maximum step size[4] of 100 A. Figure 3 shows the compressor map during the entire load cycle. The current profile is shown at the top left corner.

Figure 4 shows the oxygen excess ratio, current and compressor motor command during the 100 A tip-in. The stair-case thin line is the actual current demand. The small insert plots show the optimum values of the parameter β. In the plots, four different cases are shown: unconstrained, constrained MPC-based load governor, FRG-based load governor which uses $\tilde{O}_\infty$ (subsequently referred to as $\tilde{O}_\infty$-governor), and FRG-based load governor which uses $P \subset \tilde{O}_\infty$ (subsequently referred to as P-governor). In the unconstrained case, the surge constraint is violated during tip-outs as is shown in the compressor map plot. Also the oxygen starvation constraint is not met during the tip-in period and oxygen excess ratio almost

---

[4]A simple kinetic energy calculation shows that accelerating a 1000 kg vehicle from 20 m/s to 22 m/s (45 mph to 50 mph) in 1 second requires almost 120 A on a 350 volt BUS that connects the fuel cell with a traction motor.

reaches the critical value of 1. The constrained MPC-based load governor, $\tilde{O}_\infty$-governor and P-governor enforce all constraints. Most of the time the constrained MPC and the $\tilde{O}_\infty$-governor perform identically. They both negotiate the constraints by moving along the constraint boundary. There is a small difference between the two during tip-out transients which is due to the extra degrees of freedom of the MPC-based design. The P-governor performs more conservatively and avoids constraint violation by a certain margin. This happens because $P \subset \tilde{O}_\infty$. The advantage of the P-governor is in reduced online computation load as we further illustrate in the next section.

V. COMPUTATIONAL REQUIREMENTS

The computational requirements of each algorithm can be a deciding factor for which one gets implemented. The computational load of an algorithm may be characterized by the number of floating point operations (flops) performed. In reference [6] the number of flops (multiplication, additions, comparisons) for online calculation of fast reference governor is estimated to be $(5n_v + 2n + 4) \times m$ where n_v, n, and m are the number of reference commands, total number of states and total number of constraints. Here in this paper $n_v = 1$ and $n = 10$. For the $\tilde{O}_\infty$-governor, $m = 348$ and therefore the estimated number of flops is 10,092. For the P-governor, $m = 77$ and the number of flops reduces to 2233, an almost five fold reduction. Finding a formula for the MPC-

TABLE I

NUMBER OF ONLINE FLOATING POINT OPERATIONS FOR DIFFERENT LOAD GOVERNOR DESIGNS.

| LG Design | Flops | | | Total CPU |
	Avg.	Max	Min	time (sec)
Fast LG with $\tilde{O}_\infty$ 348 constraints 1 variable	8622	8808	8573	15.9
Fast LG with 1.2 Shrinkage 77 constraints 1 variable	2082	2118	2069	4.0
MPC 30 constraints 10 variables	20417	81870	4775	3.8

governor flops is not straightforward. At each step a quadratic program is solved based on an iterative procedure and therefore the number of flops may vary from one step to another. We used the flops command in MATLAB[5] to get an estimate for the actual number of flops for different load governor designs[6]. The average, maximum and minimum number of recorded flops is summarized in Table I. MATLAB also provides an estimate of the CPU time spent on running a selected portion of the code. The last column of Table I shows the total CPU time required for online part of the overall simulation (including plant model) on an 866 MHz Intel[7] Pentium III processor. The CPU times correspond to simulation of the model, controller and the load governor.

For the $\tilde{O}_\infty$ and P governors the number of flops given by MATLAB has little variation from one step to another and appears to be relatively close to the theoretically estimated value. The constrained MPC-governor requires a larger number of flops due to the larger number of optimization variables and constraints. Moreover due to iterative nature of QP solutions, the number of flops can vary between steps. Typically more calculations are performed by the MPC algorithm when the system is close to a constraint boundary. At the same time, Table I shows that for our simulation setup the total CPU time for the overall simulation

(including plant model) does not appear to linearly correlate with the number of flops required by the algorithms.

We note that 2082 flops at an update rate of 10 msec for the P-governor case is within the capability of automotive microcontrollers although this is still a rather large computational task. In fact, an optimized implementation in a production micro-controller simulator showed that the $\beta(k)$ can be calculated within 1.3 msec; this calculation requires close to 4 kbytes of ROM (this represents the total size of the code size and constants). The computational effort involved can be further mitigated by changing the update rate. For example, it has been shown via simulations that the update rate of 20 msec still yields acceptable performance. Further, when $\beta^* = 1$ and the reference is constant or varies slowly, then β^* can be kept equal to 1 without performing any calculation, thereby further reducing the computational load for the FRG-based load governor case.

VI. IMPLEMENTATION OF THE FRG-BASED LOAD GOVERNOR ON NONLINEAR MODEL

The load governor designs, described above, are based on the assumption that the plant is linear. In this section we apply the $\tilde{O}_\infty$-based governor to the nonlinear model of the fuel cell system. Recall that in the $\tilde{O}_\infty$-based load governor, the parameter β^* was a function of the linear state of the system, current demand, current taken from the fuel cell and $\tilde{O}_\infty$.

We first attempted to use $\tilde{O}_\infty$ to guard against constraint violation, while replacing the linear state with the nonlinear state, x_{nl}. While this approach worked well for small to medium deviations from the operating point (50 A steps in current demand up or down) it failed to perform satisfactorily for larger steps. Specifically, during a current step of 100 A, β^* was set to zero by the governor and the reference current tracking was lost. This was due to differences between linear and nonlinear systems which caused the nonlinear state to be outside of $\tilde{O}_\infty$ in steady-state.

Next we attempted to remedy this situation by adjusting the nonlinear state by the difference between linear and nonlinear steady states. Using equation (11), the linear state equilibrium for a current level I_{fc} is $x_{ss} = (I - A_{cl})^{-1}B_v I_{fc}$. The nonlinear equilibrium corresponding to I_{fc} was calculated off-line by simulation under different loads and stored in a look-up table, $\Gamma(I_{fc})$. The adjusted state is then calculated according to the formula,

$$x_{adj}(k) = x_{nl}(k) - \Gamma(I_{fc}(k)) + (I - A_{cl})^{-1}B_v I_{fc}(k). \quad (18)$$

The dynamics of the process are still predicted by the linearized model. It was shown in the simulations with nonlinear model that the load governor modified by (18) was able to reduce large excursion into the surge region during the step of 100 A and that it reduced (but failed to eliminate) oxygen starvation constraint violation of 1.9 during the first most aggressive tip-in; the load governor enforced the oxygen starvation constraint during other, less aggressive tip-ins and tip-outs. The system response, however, was jittery, since the load governor often and sporadically would set β to zero.

To improve our results, another approach was pursued wherein the difference between the nonlinear plant and its linear model were reflected in the construction of $\tilde{O}_\infty$. Specifically, we redefined O_∞ as follows:

$$O_\infty = \{(x, I_{fc}, d) : y(k) = Cx(k) + D_v I_{fc} + d \in Y \quad \forall k \in Z^+\}, \quad (19)$$

where d is a constant output disturbance term. At each instant k during online calculations, the disturbance d is the difference between the plant output and the output predicted by the linear model:

$$d(k) = y_{nl}(k) - y_{ln}(k). \quad (20)$$

The dimension of the admissible set is increased by the number of outputs n_y, i.e. $O_\infty \subset R^{n+1+n_y}$. The governor parameter β is determined as follows:

$$\beta^* = max\{\beta \in [0,1]: \begin{bmatrix} Ax + B_v I_{fc} \\ I_{fc} \\ d \end{bmatrix} + \beta \begin{bmatrix} 0 \\ I_d - I_{fc} \\ 0 \end{bmatrix} \in \tilde{O}_\infty\}, \quad (21)$$

where in (21) we use the state predicted by the linear model and account for the mismatch between the plant and the model via the disturbance term, d. This approach is intended to reduce the governor sensitivity to model uncertainty; in our case the mismatch between the linear and nonlinear models. This correction can help in eliminating constraint violation and jitter in the response. Figures 5 and 6 confirm that surge constraint violation is considerably reduced and that oxygen starvation constraint violation and jitter are eliminated. Note also that for this implementation of the governor only the state predicted by the linear model and the measurement of the output, $y_{nl}(k)$, are needed; the full plant state (i.e., state of the nonlinear model) does not need to be known.
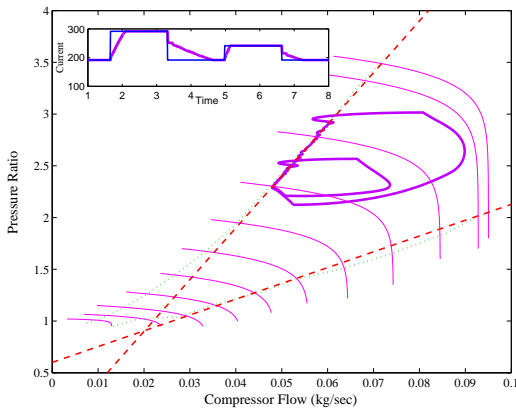

Fig. 5. The surge excursion is noticeably reduced when the disturbance observer is used.
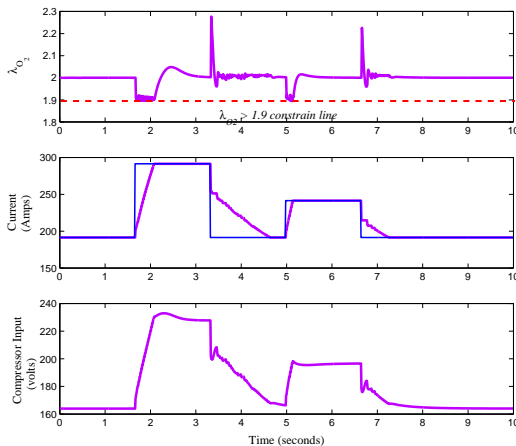

Fig. 6. Performance of the fast load governor in nonlinear simulations without state-feedback. Inclusion of the disturbance eliminates constraint violation.

VII. Conclusions

The paper explored two load governor designs for preventing oxygen starvation and compressor surge and choke on the fuel cell air supply side. The first design was based on Model Predictive Control (MPC-based load governor) applied to a linearized system model. The second design was based on the fast reference governor

approach (FRG-based load governor) applied to a linear system model. The FRG-based load governor, which is in principle suboptimal to the MPC-based load governor, was shown to perform nearly as well as the MPC-based load governor while requiring smaller on-line computational effort as measured by the number of flops. When implemented for the nonlinear plant model, the FRG-based load governor required an adjustment to compensate for differences between linear model and nonlinear system. Introducing a step disturbance observer in the load governor design allowed to nearly eliminate constraint violation and jitter in the response. The step disturbance observer eliminates the need for measurement or estimation of the plant states.

VIII. Acknowledgements

References

[1] G. Boehm, D. Wilkinson, S. Khight, R. Schamm, and N. Fletcher, "Method and appratusfor operating a fuel cell," *United States Patents 6,461,741*, 2002.
[2] J. Pukrushpan, A. Stefanopoulou, and H. Peng, "Control of fuel cell breathing," *IEEE Control Systems Magazine*, vol. 24, no. 2, pp. 30–46, April 2004.
[3] A. Vahidi, A. Stefanopoulou, and H. Peng, "Model predictive control for starvation prevention in a hybrid fuel cell system," *Proc. of American Control Conference*, pp. 834–839, 2004.
[4] Jan Tommy Gravdahl and Olav Egeland, "Centrifugal compressor surge and speed control," *IEEE Transactions of Control Systems Technology*, vol. 7, no. 5, pp. 567–579, September 1999.
[5] J. Sun and I. Kolmanovsky, "A robust load governor for fuel cell oxygen starvation protection," *Proceedings of the American Control Conference*, June 2004.
[6] E. Gilbert and I. Kolmanovsky, "Fast reference governors for systems with state and control constraints and disturbance inputs," *International Journal of Robust and Nonlinear Control*, vol. 9, pp. 1117–1141, 1999.
[7] J. Pukrushpan, A. Stefanopoulou, and H. Peng, *Control of Fuel Cell Power Systems: Principles, Modeling, Analysis, and Feedback Design*, Springer-Verlag, London, UK, 2004.
[8] J.A. Adams, W-C Yang, K.A. Oglesby, and K.D. Osborne, "The development of fords p2000 fuel cell vehicle," *SAE Paper 2000-01-1061*, 2000.