Iterative optimization of the filling phase of wet clutches

Bruno Depraetere¹, Gregory Pinte², Jan Swevers¹
¹ Department of Mechanical Engineering, Division PMA, Katholieke Universiteit Leuven
² Flanders’ Mechatronics Technology Centre
Celestijnenlaan 300B¹ / 300D², 3001 Heverlee, Belgium

Abstract—This paper considers the control of wet clutches, and presents a two-level control strategy to learn and adapt the control signals during normal machine operation. With this approach it is possible to avoid the current practise of experimental calibrations, where regular recalibrations are needed to compensate for time-varying dynamics, e.g. due to wear and changes in oil temperature. On a low level, the developed controller determines the actuator signal by solving an optimal control problem before each engagement of the clutch. The models and constraints for this optimization problem are iteratively updated by a high-level controller, which consists of a recursive identification algorithm to model the system dynamics, and of an ILC-type algorithm to learn appropriate values for the constraints. The performance and robustness of this control scheme are validated on an experimental test setup.

I. INTRODUCTION

Wet clutches are devices to transmit torque from one shaft to another by means of friction. A cross-section of a clutch is schematically drawn in figure 1. It contains two sets of friction plates, one that can slide in grooves on the inside of the drum, and another that can slide in grooves on the outgoing shaft. Torque can be transferred between the shafts by pressing both sets together with a hydraulic piston, which can be realized by sending an appropriate control signal to the servovalve in the line to the clutch. Initially, during the filling phase, the clutch chamber fills up with oil and the pressure builds up, until it is high enough to compress the return spring and accelerate the piston. When the piston advances far enough and presses the plates together, the filling phase ends and the slip phase begins. During the slip phase torque is transferred, causing the difference in rotation speeds between the shafts, called the slip, to decrease, until both shafts have the same rotation speed.

Wet clutches are commonly used in automatic transmissions for off-highway vehicles and agricultural machines to transfer torque from the engine to the load. By disengaging one clutch and engaging another, different transmission ratios can be realized. When a clutch engagement is requested, an operator expects a fast response without vibrations. Torque transfer should thus commence as soon as possible without introducing torque discontinuities and peaks, which can be realized by a short filling phase followed by a smooth transition into the slip phase.

The two main challenges for wet clutch control are (i) the complex, non-linear behavior, with a sudden change in dynamics when the piston comes into contact with the plates, and (ii) the variation of these dynamics over time, due to changes in oil temperature and wear. The first challenge is addressed in [1] and [2] by using separate controllers for each phase: first, feedforward actions are used to bring the clutch into the slip phase, followed by feedback controllers to regulate the slip or the pressure respectively, [3] and [4] developed complex models for the whole engagement process, and applied these to design non-linear feedback [3] or feedforward controllers [4]. For good operation, all these techniques require a considerable modeling effort. Furthermore, they do not consider the time-variation of the dynamics, which makes robust wet clutch control a challenging problem [5]. Industrial clutch controllers cope with these issues by using parameterized feedforward signals and by regularly recalibrating the signal parameters during machine servicing. In an attempt to avoid this downtime, various patents have been claimed to derive empirical rules for adjusting the signal parameters during normal machine operation, based on observations of past engagements [6], [7], [8]. However, no systematic framework is available.

In this paper a two-level robust clutch control strategy is proposed that learns and adapts the control signals during normal machine operation, in order to maintain performance despite variable operating conditions and without recalibrations. The presented two-level control scheme is shown in figure 2. On a low level, the controller derives the servovalve current for the filling phase by numerically solving a constrained optimization problem in between engagements. The last value of this control signal is maintained during the slip phase, as shown in figure 3, so a constant chamber pressure is obtained, pressing the plates together with a constant force. This results in a constant torque transfer and a linear acceleration of the load, ensuring operator comfort by avoiding torque peaks. On a higher level, a learning algorithm exploits the repetitiveness of the engagements by iteratively updating the model and learning appropriate values for the constraints used by the optimization problem. The model parameters are updated using a recursive identification algorithm, while the setpoints

Fig. 1. Cross-section of a wet clutch and its components
for the piston position and pressure are updated using an ILC-type learning algorithm (Iterative Learning Control), based on a criterion to quantitatively assess the engagement quality.

The rest of this paper is organized as follows. Sections II and III discuss the low and high-level controllers respectively. Section IV discusses the experimental validation of the developed control strategy. Finally, section V presents some conclusions and suggestions for future research.

II. LOW LEVEL: CONTROL SIGNAL OPTIMIZATION

The servovalve current for the filling phase is numerically optimized before each engagement. This is implemented by minimizing $f_{\text{final}}$, the time needed to reach the state specified by the setpoints for piston position $z$ and pressure $p$, which are provided by the high-level controller. The setpoint $p_{\text{final}}$ for the pressure needs to be reached with $p = 0$, so a constant torque can be obtained. On the other hand, the setpoint $z_{\text{final}}$ for the piston position needs to be reached with a low piston velocity, $\dot{z} < \epsilon$, so the piston makes contact with the plates at a low velocity. In the optimization problem the pressure and piston position are predicted as a function of the servovalve current $u$, using a model which is also provided by the high-level controller.

The optimization problem is solved in discrete-time for a given sampling time $T_s$. Minimizing $f_{\text{final}}$ thus corresponds to minimizing the number of samples $K$ before the final state is reached. Since minimizing $K$ is a non-convex problem, it is reformulated as a feasibility search over a series of convex subproblems [9], each for a different number of samples $K$. Using a bisection algorithm, the lowest feasible value $K^*$ and the corresponding control signal can then be found by solving a limited number of feasibility problems, given by:

$$\begin{align*}
\min_{u(:),x(:)} & \sum_{k=0}^{K-2} \frac{|u(k+1) - u(k)|}{T_s} , \\
\text{s.t.} & x(0) = x_0 , \\
& x(k+1) = Ax(k) + Bu(k), k = 0, \ldots, K-2, \\
& p_{\text{final}} = p(K-1), \\
& z_{\text{final}} = z(K-1) - z(K-2) < \epsilon T_s, \\
& u_{ib} < u(k) < u_{ub}, \\
& y_{ib} < Cx(k) + Du(k) < y_{ub}, \\
& \text{for } k = 0, \ldots, K-1, \\
\end{align*}$$

(1a)

where (1a) is a cost function minimizing the $L_1$-norm of the derivative of the control signal, which is added to ensure the smoothest control signal $u$ is found if the problem is feasible. This is needed because only integer values of $K$ are allowed, so many different control signals $u$ can exist for the optimal value $K^*$. Eqs. (1b) to (1c) express the dynamics of the system, using matrices $A, B, C$ and $D$ of the state-space model relating input $u$ to the two outputs $z$ and $p$. The end-point conditions are given in (1d) to (1f). (1d) specifies the setpoints for the pressure and piston position to be reached at $t_{\text{final}}$, while (1e) and (1f) specify the constraints on their derivatives, $p = 0$ and $\dot{z} < \epsilon$ respectively, with $\epsilon$ a small, positive number. (1g) and (1h) are the lower and upper bounds for the inputs and outputs respectively, given by the physical limitations of the system.

Because (1) is a linear problem, the non-convex problem of finding the time-optimal control signal $u$ has been reduced to solving a series of linear feasibility problems.

III. HIGH LEVEL: MODEL AND SETPOINT LEARNING

This section describes the algorithms for the high-level learning of the model and the setpoints that are needed for the optimization problem. First, the recursive identification of the pressure model is discussed in III-A, followed by the algorithm for updating the pressure setpoint in III-B. The learning of the model and the setpoint for the piston position is discussed next, in III-C.

A. Recursive identification of pressure model

The low-level optimization problem requires a model relating the servovalve current to the pressure. Pressure measurements in the line to the clutch are readily available in industrial transmissions, so these can be used to identify a model for the filling phase. A recursive technique is developed, which offers the possibility to improve the model accuracy and to compensate for slowly varying dynamics by combining information from multiple engagements. Since only the filling phase needs to be modeled, a linear, low order, discrete-time model is used. Because the behavior is considered time-invariant over the span of a single engagement, the model parameters are updated once after each engagement, using the complete batch of measured data. This differs from the traditional recursive estimation techniques where updates are calculated after each sample [10]. Once the model parameters are estimated, the matrices $A, B, C$ and $D$ of the state-space model (1) are updated accordingly.

The recursive estimation algorithm will first be derived, before its relation to other recursive estimators is briefly discussed.
The system is modeled as a discrete-time output error model [10], with dynamics described by
\[ y(k) = \frac{B(q^{-1}, \theta)}{F(q^{-1}, \theta)} u(k-n_k) + e(k), \]  
(2)
where \( u(k) \) and \( y(k) \) are the input (current) and output (response) respectively, \( e(k) \) is measurement noise and \( n_k \) is the number of delays in the system. The delay operator is denoted by \( q^{-1} \), and
\[ B(q^{-1}, \theta) = b_1 + b_2 q^{-1} + \ldots + b_n q^{-n_b}, \]  
(3)
and
\[ F(q^{-1}, \theta) = 1 + f_1 q^{-1} + \ldots + f_n q^{-n_f}. \]  
(4)
are polynomials of \( q^{-1} \) that depend on the parameter vector \( \theta = [b_1 b_2 \ldots b_{n_b} f_1 f_2 \ldots f_{n_f}]^T \).

First, consider a single batch of measured data \( y = [y(1) y(2) \ldots y(N)]^T \) and \( u = [u(1) u(2) \ldots u(N)]^T \). For the given model structure, the output can be predicted as
\[ \hat{y}(k|\theta) = \frac{B(q^{-1}, \theta)}{F(q^{-1}, \theta)} u(k-n_k) = \varphi(k, \theta)^T \theta, \]  
(5)
where \( \varphi(k, \theta) \) is defined as
\[ \varphi(k, \theta) = [u(k-n_k) u(k-n_k-1) u(k-n_k-n_b+1) \ldots u(k-n_k-n_f)]^T. \]  
(6)
An estimate for the system parameters can now be found in the weighted least squares sense as
\[ \theta^* = \arg \min_{\theta \in D_{\theta}} (y - \Phi(\theta))^T W (y - \Phi(\theta))^T \theta, \]  
(7)
where \( W \) is a positive definite weighting matrix, \( \Phi(\theta) = [\varphi(1, \theta) \varphi(2, \theta) \ldots \varphi(N, \theta)] \) and \( D_{\theta} \) is the set of stable models. This is a non-linear, non-convex least squares problem, which can be solved as a sequential quadratic program (SQP)[11]. In this SQP, (7) is approximated at the current estimate \( \hat{\theta} \) by considering \( \Phi(\theta) \) fixed and independent of \( \theta \), denoted by \( \Phi_0 \). Next, a new estimate \( \hat{\theta} + \alpha \delta \theta \) is calculated, where \( \delta \theta \) is the update direction given by
\[ \delta \theta = \arg \min_{\delta \theta} (y - \Phi_0(\hat{\theta} + \delta \theta))^T W (y - \Phi_0(\hat{\theta} + \delta \theta))^T \theta, \]  
(8)
\[ = \arg \min_{\delta \theta} \delta \theta^T \Phi_0 W \Phi_0^T \delta \theta - 2 \delta \theta^T \Phi_0 W (y - \Phi_0(\hat{\theta})), \]  
(9)
\[ = (\Phi_0 W \Phi_0^T)^{-1} \Phi_0 W (y - \Phi(\hat{\theta})), \]  
(10)
and \( \alpha \) is a scalar step size selected using backtracking2. This iterative estimation process is repeated by calculating new estimates for \( \hat{\theta} + \alpha \delta \theta \), until it converges and the solution is found. In a non-recursive framework, a good initial estimate is needed to guarantee convergence to the global optimum. In the considered recursive case however, the model estimated after the previous engagement is used as the starting point for the SQP.

As the SQP converges, the updates \( \delta \theta \) converge to 0 and so does \( \Phi_0 W (y - \Phi(\hat{\theta})) \) in (10). As a result, around the optimal solution, this term can be neglected and the cost function in (9) can be approximated using only the quadratic term.

Denote this first batch as \( i \), with for example \( y_i \) and \( \hat{y}_i \) vectors of the measured and predicted outputs. When adding a second batch of data, \( i+1 \), the estimation problem yields
\[ \hat{\theta}_{i+1} = \arg \min_{\theta \in D_{\theta}} \{(y_{i+1} - \hat{y}_{i+1}(\theta))^T W (y_{i+1} - \hat{y}_{i+1}(\theta)) \} \]  
\[ + (y_i - \hat{y}_i(\theta))^T W (y_i - \hat{y}_i(\theta)), \]  
(11)
In this equation, \( \hat{\theta}_{i+1} \) is the estimate for \( \theta \), made after the \( (i+1) \)-th batch is added and using data from batches \( i \) and \( i+1 \). The second term in this equation is the same as (7), and can thus be approximated by its quadratic approximation when a model \( \hat{\theta}_i \) for the first batch is known. Substituting the first term of (9) into (11) and using the notation \( H_i = \Phi_\theta W \Phi_\theta^T \), yields
\[ \hat{\theta}_{i+1} = \arg \min_{\theta \in D_{\theta}} \{(y_{i+1} - \hat{y}_{i+1}(\theta))^T W (y_{i+1} - \hat{y}_{i+1}(\theta)) \} \]  
\[ + (\theta - \hat{\theta}_i)^T H_i (\theta - \hat{\theta}_i), \]  
(12)
which is again a non-convex, non-linear least squares problem. Using a similar SQP approach, the update direction for a current estimate \( \hat{\theta} \) is given by
\[ \delta \theta = \arg \min_{\delta \theta} \delta \theta^T (\Phi_\theta W \Phi_\theta^T + H_i) \delta \theta \]  
\[ - 2 \delta \theta^T (\Phi_\theta W (y - \Phi_\theta^T \hat{\theta}) - H_i (\hat{\theta})) \]  
(13)
\[ = (\Phi_\theta W \Phi_\theta^T + H_i)^{-1} \left( \Phi_\theta W (y - \Phi_\theta^T \hat{\theta}) - H_i (\hat{\theta}) \right). \]  
(14)
When this SQP is solved the series of \( \delta \theta \) again converges to 0, and so does
\[ \Phi_\theta W (y_{i+1} - \Phi_\theta^T \hat{\theta}) - H_i (\hat{\theta}) \rightarrow 0. \]  
(15)
Once the optimal solution is found, the cost function for the 2 batches can thus approximated by the first, quadratic, term in (13). When a third batch of data is added, the new estimate can be found in the same way as in (12), requiring only \( H_{i+1} - \Phi_{\theta_{i+1}} W \Phi_{\theta_{i+1}}^T + H_i \) and the model \( \hat{\theta}_{i+1} \) from the first two batches. Based on this reasoning, a recursive estimation algorithm can thus be derived, which consists of solving the SQP (12) each time a batch of data becomes available, and storing \( H \) and \( \hat{\theta} \) for future calculations. A minor modification is still needed when the system behavior varies with time: Past data is weighted with a forgetting factor, which is realized by multiplying the second term in (12) with a positive scalar \( \alpha \). The derived algorithm can be compared to the standard recursive least squares estimator [10], when it calculates an estimate after each batch instead of after each data sample. The main difference lies in the fact that the standard recursive least squares estimator performs only one iteration each time a new batch of data becomes available, while the presented SQP approach performs several iterations until convergence is obtained. Using the SQP is justified since solving several iterations for each batch increases the convergence speed, and sufficient computational time is available in between engagements.

1The notation of [10] will be used throughout this section.

2The normal backtracking algorithm [11] is adapted to make sure that the model remains stable, reducing the step size until either a stable model is obtained or the step size becomes too small and the SQP is stopped.
B. Iterative learning of pressure setpoint

To ensure that the optimized control signal brings the piston to a position close to the friction plates and the pressure to a level such that a smooth transition into the slip phase is possible, the low-level optimization problem requires an appropriate value for \( P_{\text{final}} \) in the constraint (1d). An ILC-type algorithm is therefore designed to learn this value, based on a quantitative indicator of engagement quality, which is estimated after each engagement. For this indicator the time \( t_{\text{sync}} \) is selected. This is the moment when the slip becomes negligible and both shafts rotate at the same speed, and can easily be estimated since both rotation speeds are measured. The value of \( t_{\text{sync}} \) depends strongly on the amount of torque transferred during the slip phase, which is in turn closely related to the pressure pressing the plates together. A learning controller that iteratively adapts the pressure setpoint can thus be used in an attempt to match \( t_{\text{sync}} \) to a given reference \( t_{\text{sync}}^* \). Throughout this paper \( t_{\text{sync}}^* \) is always set to 1s after the start of torque transfer, giving the load 1s to accelerate, independent of the load. A new value for the setpoint is calculated after each engagement, according to

\[
P_{\text{final}, i+1} = P_{\text{final}, i} + \gamma i (t_{\text{sync}}^* - t_{\text{sync}}),
\]

(16)

which is an ILC-like learning algorithm, with \( \gamma i \) a saturating proportional controller, shown in figure 4. After completing an engagement and estimating \( t_{\text{sync}} \), this updated value for the pressure setpoint is then inserted into (1d).

C. Iterative learning the position constraints

Optimization problem (1) requires a setpoint for the piston position, as well as a model relating the servovalve current \( u \) to the piston position \( z \). The position setpoint should ideally not change much from engagement to engagement, as the travel distance only varies very slowly due to wear. The dynamic behavior of the piston on the other hand varies significantly, due to the viscosity depending strongly upon the oil temperature. To maintain engagement quality under all operating conditions, it is thus needed to iteratively update the position model and compensate for this variation. Unfortunately, no position measurements are available in industrial transmissions and also not on the test-setup which will be used in section IV to validate the controllers. A recursive identification algorithm similar to that of section III-A can therefore not be applied. Instead, a linear, first-order, low-pass model with a cutoff-frequency of 1Hz is used to express the piston position as a function of the pressure. This model is available from experiments on a previous test-setup, and gives a crude approximation of the relation between pressure and position. To account for the uncertainty and variation in this model, it is extended by adding a single unknown scalar multiplier \( \gamma \), and this parameter is iteratively adapted by the high-level controller. In the optimization problem (1), the setpoint \( z_{\text{final}} \) is thus left intact, while the model for the position is rescaled by changing the corresponding coefficients in \( C \) and \( D \).

The high-level controller adapts the multiplier \( \gamma \) based on an estimate of the start of the slip phase, \( t_{\text{start}} \). This moment can be determined by detecting the start of torque transfer, when the output rotation speed increases and the input rotation speed decreases. The value of \( t_{\text{start}} \) is a good measure for whether the optimized control signal brings the piston close to the friction plates, because the slip phase commences when the piston first makes contact with the plates. The goal is to start the slip phase as soon as possible, which is implemented by selecting a reference \( t_{\text{start}}^* \) equal to \( t_{\text{final}} + 50 \text{ms} \), so the slip phase should start 50ms after the optimized control signal ends. Because \( t_{\text{final}} \) is itself always minimized in (1), this effectively also minimizes the time needed to reach the slip phase and start torque transfer.

A similar update law as (16) for the pressure is used, trying to get \( t_{\text{start}} \) to converge towards \( t_{\text{start}}^* \): It is however applied on \( \frac{1}{\gamma} \) instead of on \( \gamma \), which yields the following update law

\[
\frac{1}{\gamma} = \frac{1}{\gamma} + \gamma_* \left( t_{\text{start}}^* - t_{\text{start}} \right),
\]

(17)

where \( \gamma_* \) is similar to \( \gamma_i \), and is also shown in figure 4. Using \( \frac{1}{\gamma} \) instead of \( \gamma \) makes the interpretation of the learning process easier, as \( \frac{1}{\gamma} \) can be seen as a measure for the travel distance. When \( t_{\text{start}}^* - t_{\text{start}} > 0 \), the piston advances too far and bumps into the plates too soon, which means that \( \frac{1}{\gamma} \) is too large and needs to be reduced. When \( t_{\text{start}}^* - t_{\text{start}} < 0 \), the piston stops too far from the plates, hence \( \frac{1}{\gamma} \) is too small and needs to be increased. When however \( \frac{1}{\gamma} \) is chosen correctly, the predicted position resembles the real position and \( t_{\text{start}} \) corresponds well to the target \( t_{\text{start}}^* \).

Note that when wear affects the travel distance, the algorithm will continue to work, since this will cause \( t_{\text{start}} \) to deviate from its target and (17) will compensate for this.

IV. EXPERIMENTAL RESULTS

The developed control strategy is validated on the experimental test bench shown in figure 5. The SOHB TE10 transmission to be controlled is located on the left, driven by an induction motor (30 kW). A secondary SOHB RT20000 transmission and a flywheel (1 kgm²) are used to vary the load observed by the controlled transmission. In the experiments the first gear of the controlled transmission will be engaged, with the input shaft driven at a constant speed of 1500 rpm. The output shaft always starts at 0 rpm before the clutch is engaged with different loads and at different oil temperatures. For the recursive identification, model structure (2) is used with \( n_p = 1 \), \( n_f = 3 \) and \( n_q = 2 \). The calculations are performed in MATLAB, using MLIB to communicate with a dSPACE 1103 control board. A sensor measuring the transferred torque is present on the setup.
but it is only used to illustrate the engagement quality and not for the controllers, as this type of sensor is not readily available in normal transmissions. A temperature sensor and a cooling system are also present, which are used to keep the oil temperature steady. The only sensors that are used for clutch control are pressure gages measuring the clutch pressures and incremental encoders measuring the shaft rotation speeds.

A first set of experiments has been performed with a fixed load at a constant oil temperature of 40°C. Suboptimal initial setpoints were chosen and the model was initialized by a short feedforward run. After this initialization the presented controller was activated, and a first control signal was optimized using the initial setpoints and model. The results for the learning process are shown in figures 6 up to 9. In figure 6 the predicted and measured pressures are compared for the 1st, 4th and 16th iterations, focusing on the transient at the end of the filling phase. At this time the pressure drops down from its initial high value and settles to its stationary value to be maintained throughout the slip phase. The figure shows that when more iterations pass the predictions improve and finally match the measurements closely. At the same time, the pressure setpoint and position model are also being adapted by the learning controllers, as shown in figure 7. They initially change quite significantly before converging, after about 15 – 20 engagements. This means that the quality indicators are also approaching their targets, as can be verified in figure 8. It follows from this figure that initially torque transfer starts too soon and the load accelerates too quickly. The setpoints are adapted to bring these indicators to their respective targets, until the performance specifications are met and comfortable engagements are obtained. This convergence process is also illustrated in figure 9, showing the current and torque during the 1st, 4th and 16th iteration in (a) and (b) respectively. In (a), the first control signal was optimized using a pressure setpoint that was too high, a position model that caused the piston to travel further than desired and a pressure model with good steady state behavior but that fails to predict the transient behavior correctly. As a result, the piston bumped into the friction plates at a high velocity causing the torque to start too soon and the load to accelerate too quickly, as observed earlier. This also leads to a high torque peak causing discomfort for the operator. Over the course of the following engagements the setpoints and models for the optimization problem are adapted resulting in different control signals for each engagement, leading to torque profiles that are increasingly smooth. After convergence, the initial steep increase in torque is gone and torque transfer starts after about 350ms. The duration of the following slip phase has also converged to the desired 1s. As a result, smooth engagements are obtained, with a fast response but without torque peaks. In this experiment the convergence is quite slow, but it can be sped up by learning more aggressively when the performance is very poor. In practise however, better initial values can be chosen and then convergence is not an issue.

To validate the robustness, other sets of experiments have been performed, one at 70°C with the same load, and another at 40°C with a higher load. For these experiments, the results after convergence are shown in figure 10. At 70°C the clutch fills up more easily than at 40°C due to the lower oil viscosity. The high-level controller compensates for this by rescaling the position model. The optimization problem recalculates the control signal using this rescaled model, yielding that the current drops down more quickly than at 40°C, but the resulting torque profile does remain similar and still matches the specifications. When the load is increased but the temperature is kept at 40°C, the constant servovalve current at the end of the optimized part increases. This is due to an
increase in the pressure setpoint by the high-level controller, to ensure the heavier load can be accelerated in the same period of 1s. The torque profile for this case is also very smooth and similar, and matches the specifications, despite the larger values of transferred torque.

V. CONCLUSIONS AND FUTURE WORK

This paper presents a two-level, optimization-based control strategy for wet clutches. At a high level, the system behavior during the filling phase is recursively modeled and setpoints are iteratively learned using indicators of engagement quality. This data is used to formulate the constraints of an optimal control problem, which is solved numerically by the low-level controller to determine the control signal for the next clutch engagement. This iterative procedure adapts to changes in oil temperature and load has been demonstrated.

Future work will concentrate on extending the model towards the slip phase, so the entire control signal can be optimized. The main challenge lies in an accurate detection of the transition between the filling and slip phases, as well as smooth control over this transition.

ACKNOWLEDGMENT

This work has been carried out within the framework of projects IWT-SBO 80032 of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) and G.0422.08 of the Research Foundation – Flanders (FWO - Vlaanderen). This work also benefits from K.U.Leuven-BOF EF/05/006 Center-of-Excellence Optimization in Engineering and from the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office.

REFERENCES