

# Optimization aided Loop Shaping for Motion Systems

Dennis Bruijnen, René van de Molengraft and Maarten Steinbuch

**Abstract**—An approach is proposed which improves the quality and speed of manual loop shaping. Loop shaping is an iterative and creative controller design procedure where the control engineer uses frequency response function (FRF) data of the plant to shape the open loop response such that it satisfies stability, performance and robustness specifications. The advantage compared to automated controller design methods is that the control engineer can exploit all available a priori knowledge and expertise about the plant during the design process. As an assisting tool in manual loop shaping, we add a global optimization method, i.e. a genetic algorithm, where the objective function resembles as good as possible what the control engineer wants. As a result, the tuning process is substantially accelerated. The approach has been implemented in a Matlab-based control tuning tool showing good results.

## I. INTRODUCTION

Loop shaping in the frequency domain [4], [11] is a widely used method by control engineers in practice. It is an iterative and creative process to achieve desired closed loop behavior in terms of stability, performance and robustness to model uncertainties and disturbances. The main advantages of frequency loop shaping are freedom of choice of the controller structure and the possibility for the control engineer to assess his requirements directly after each adjustment to the controller. On the other hand, a lot of optimization based methods exist that automate the controller design process, such as LQ, LQG [1],  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  [3]. Ideally, frequency loop shaping could be replaced totally by these methods. However, the requirements of the control engineer always have to be compromised due to the limitations each method has, such as the class of controllers which can be handled, the performance criteria which can be optimized and to what extend disturbance models can be incorporated. The result is a suboptimal controller in view of the actual control engineer's requirements. Therefore, frequency loop shaping remains a popular, often applied method, in particular for SISO systems.

Automatic loop shaping is an active field of research [2], [5], [8], [13], [16]. The main challenge is to formulate the objective function for optimization such that it resembles the requirements of the control engineer as good as possible. Often, and certainly in the case of fixed order or structured controllers, this results in a NP-hard optimization problem where the optimal solution can be approximated by global optimization methods, e.g. Monte-Carlo, simulated annealing [9], evolutionary algorithms, such as genetic algorithms [6], swarm-based optimization algorithms, such as particle swarm

optimization [7] etc. These algorithms are all computationally expensive and therefore they most often do not come up with the global optimal solution within an acceptable time span. Nevertheless, the suboptimal solution found can be good enough.

This paper exploits the synergy of manual frequency loop shaping and optimization aid via a genetic algorithm such that the control engineer is able to quickly design a high performance controller. The optimization aid is seen as an assisting tool to speed up the iteration process and not as a replacement for manual loop shaping. This approach enables the control engineer to use its expertise *during* the semi-automated iteration process. A software tool has been written in Matlab for SISO controller design which enables the user to: select plant data which can be a SISO model, SISO FRF-data or multiple SISO FRF-data, choose a set of filters for the controller, optimize a selection of them where some key parameters for high gain feedback loop shaping can be set, carry out the loop shaping iteration process with optimization aid until an acceptable result is obtained and finally export the resulting controller to a discretized Simulink block ready-to-use for simulations and experiments.

First, high gain feedback loop shaping is described in short. After that, the optimization aid is presented. Next, the loop shaping tool is presented and a loop shaping example with this tool is given. Finally, conclusions are drawn.

## II. HIGH GAIN FEEDBACK LOOP SHAPING

A standard feedback control loop is shown in Fig. 1. For SISO LTI systems, the transfer function from reference  $r(t)$  to output  $y(t)$ , i.e. the complementary sensitivity function, is

$$T(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} \quad (1)$$

where  $Y(s)$  and  $R(s)$  are the Laplace transforms of  $y(t)$  and  $r(t)$  respectively,  $P(s)$  is the plant model and  $C(s)$  is the controller model.

From now on we only consider steady state relations, i.e.  $s = j\omega$ . The idea of high gain feedback is to increase the loop gain  $P(j\omega)C(j\omega)$  such that  $y(t)$  resembles  $r(t)$ . From (1) it is easy to see that  $\frac{Y(j\omega)}{R(j\omega)} \approx 1$  and thus  $\frac{y(t)}{r(t)} \approx 1$  for  $|P(j\omega)C(j\omega)| \gg 1$ .

In practice, the allowable loop gain is limited by e.g. the limited actuator capacity, parasitic plant dynamics and loop delay due to digital implementation. Moreover, the loop gain is always decreased further to obtain robustness to plant uncertainties and better disturbances attenuation above the bandwidth.

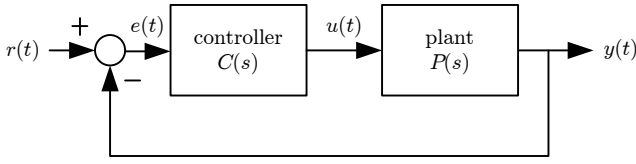


Fig. 1. Scheme of a standard feedback control loop

### A. REQUIREMENTS OF THE CONTROL ENGINEER

Define the bandwidth  $\omega_{bw}$  as the frequency where  $|P(j\omega)C(j\omega)|$  crosses the 0 dB level the first time.

$$\omega_{bw} = \inf_{\omega} |P(j\omega)C(j\omega)| \leq 1 \quad (2)$$

Below the bandwidth we have  $|P(j\omega)C(j\omega)| > 1$ , hence  $\frac{y(t)}{r(t)} \approx 1$ . If most of the frequency content of the reference signal is in this region, the reference can be tracked well using this feedback controller. However, in practice, tracking performance for motion systems is primarily determined by feedforward design. The main goal of a feedback controller design is disturbance rejection which can be achieved by high gain feedback implying a high bandwidth.

Besides this performance specification, the control engineer requires stability and robustness to plant uncertainties. For non-parametric FRF-data, the Nyquist stability criterion [4] can be applied to guarantee stability of the closed loop system. Robustness against plant uncertainties is imposed by limiting the minimal distance to the point -1 in the Nyquist plot which is equal to limiting the maximum magnitude of the sensitivity function

$$S(s) = \frac{1}{1 + P(s)C(s)} \quad (3)$$

This maximum magnitude of the sensitivity function, also called the modulus margin [10], is

$$\sup_{\omega} |S(j\omega)| \leq \alpha \quad (4)$$

where  $\alpha = 6$  dB is a common choice. For  $\alpha = 6$  dB, the modulus margin also implies a phase margin of  $30^\circ$  and a gain margin of 6 dB. However, the opposite is not true so this makes the modulus margin the most important margin for controller design.

### B. OBJECTIVE FUNCTION FOR LOOP SHAPING

For optimization, we want to have an objective function which resembles the requirements of the control engineer as good as possible. The available data of the plant is assumed to be (non-parametric) FRF-data of a stable SISO system. The proposed objective function implements the following criteria

- criterion 1: a target bandwidth
- criterion 2: low gain above the bandwidth
- criterion 3: high gain below the bandwidth
- criterion 4: constraint on the gain, phase and modulus margin

- criterion 5: constraint that the open loop should cross the 0 dB once

The constraint on the phase margin and criterion 5 are sufficient conditions to imply stability. The constraint on the modulus margin implies robustness against plant uncertainty. The gain margin is not necessary for optimization, nevertheless, it is a commonly used criterion by the control engineer and therefore it is added to the constraints.

The objective function has to combine all criteria into one value indicating the quality of an arbitrary controller. Criterion 1 can be implemented by a quadratic objective function with the minimum at  $\omega_{target}$

$$J_{bw} = w_1 (\omega_{bw} - \omega_{target})^2 \quad (5)$$

where  $w_1$  is a weighting factor.

To stimulate low gain for all frequencies above the bandwidth (criterion 2), the mean gain in dB is added to the objective function. Similarly, the mean gain in dB is added for frequencies below the bandwidth which accounts for criterion 3.

$$J_{gain} = w_2 \sum_{\omega > \omega_{bw}} 20 \log |P(j\omega)C(j\omega)| - w_3 \sum_{\omega \leq \omega_{bw}} 20 \log |P(j\omega)C(j\omega)| \quad (6)$$

where  $w_2$  and  $w_3$  are weighting factors.

The choice for  $w_1$ ,  $w_2$  and  $w_3$  is crucial. From a control engineer's point of view, it is logical to first obtain approximately the required bandwidth and after that increase the gain below the bandwidth and decrease the gain above the bandwidth, hence the weighting factor of the bandwidth criterion should be relatively large if the current bandwidth is not close to the target bandwidth. A proper choice for  $w_1$  is

$$w_1 = \frac{1}{\omega_{target}^2} \quad (7)$$

which normalizes the bandwidth criterion such that  $J_{bw} = 0.01$  for 10% bandwidth deviation.  $w_2$  and  $w_3$  are chosen to be  $\frac{1}{2000}$  such that the contribution to  $J_{gain}$  for a mean value of 20 dB, which is the order of magnitude, is also 0.01. Because criterion 2 and 3 are linear and criterion 1 is quadratic, larger deviations of the bandwidth implies  $J_{bw} \gg J_{gain}$ , hence the bandwidth criterion is dominant. Constraints (criterion 4 and 5) are incorporated in the objective function by returning a large penalty  $J_{penalty} = 10^{10}$  plus the amount of constraint overshoot  $J_{overshoot}$  which can be for example the modulus margin. The total objective function  $J$  becomes

$$J = \begin{cases} J_{bw} + J_{gain} & \text{if all constraints are satisfied} \\ J_{penalty} + J_{overshoot} & \text{if at least one constraint is not satisfied} \end{cases} \quad (8)$$

### C. OPTIMIZATION AID

The objective function, as described in the previous section, is a nonlinear function for which the global optimum solution is difficult to find when using non-parametric FRF-data. A genetic algorithm [6] is chosen to deal with this problem because the effectiveness of the method has often been shown in literature [14], [15]. In short, it is an evolutionary algorithm which searches for the optimum solution by mimicking the evolution process in nature. A population of solutions evolves where better solution have more chance to reproduce itself using mutation and crossover operations. This way, solutions can improve each generation until an adequate solution is found. The implementation of the Genetic Algorithm and Direct Search Toolbox of Matlab [12] is used. Details of how the parameters of the genetic algorithm are chosen will not be further discussed.

Advantages of this optimization method are

- freedom in choosing the objective function. The objective function can be chosen to be an arbitrary function of all criteria such that the output value represents the performance of the controller well.
- freedom of choosing the controller structure and its parameters with its bounds. The parameters can be chosen such that they have physical meaning which can be easily interpreted by the control engineer. Furthermore, the control engineer can easily fix some parameters and optimize only the rest which is a powerful method to include knowledge and expertise of the control engineer into the controller design.
- it explores the specified parameter space where it can explore solutions that the control engineer did not immediately think of.

Because in general a lot of function evaluations ( $>10000$ ) are needed for global optimization methods to obtain a good solution, it is essential that the calculation time of the objective function is as short as possible. For the objective function as described in the previous section, subjected to FRF-data, tests have shown that the amount of function evaluations lies between 100 and 1000 per second depending on the controller complexity and the frequency gridding. The result is that the optimization method for the examples in this paper comes up with a good solution in less than a minute which is considered fast with respect to manually iterating a set of controller parameters.

Moreover, for our goal it is not essential that after a single run of the algorithm the global optimal solution is found. The optimization is used as a tool which helps improving the controller during the loop shaping iteration process. During the optimization, the control engineer can observe the performance of the controller and stop the optimization at any time. Combined with the control engineer expertise, convergence to the optimal controller is accelerated.

### III. LOOP SHAPING STRATEGY

In this section we will compare classical loop shaping to loop shaping combined with an optimization tool.

### A. CLASSICAL LOOP SHAPING

In classical loop shaping, the control engineer performs the following steps:

- 1) Measure the FRF from an experimental setup or extract FRF-data from a model by computing  $P(j\omega)$  for a selection of values for  $\omega$ .
- 2) Select a controller structure
- 3) Stabilize the control loop
- 4) Increase the bandwidth by changing the parameters of the controller until one of the constraints has been reached
- 5) If the controller is not yet good enough, add a controller block and go back to step 3
- 6) Validate the controller performance with simulations and experiments

The expertise of the control engineer, which allows him to use a priori knowledge in the design process, determines the quality of the final resulting controller. In practice, step 3 and 4 are time consuming and have to be repeated several times to obtain a satisfying result.

### B. OPTIMIZATION AIDED LOOP SHAPING

With respect to classical loop shaping, optimization aid enables the following features

#### 1) OPTIMIZING CONTROLLER PARAMETERS:

If there are only a few parameters it is easy for the control engineer to tune the controller using the rules of thumb at hand. However, if the controller complexity increases, it is not that straightforward anymore how to manually tune the controller. In that case, an optimization algorithm can be much faster in improving the controller parameters. A first guess of how the controller should look like can be obtained by specifying the controller blocks, the desired bandwidth and the gain, phase and modulus margin and then run the optimization algorithm. The optimization process can be interrupted by the user at any time, after which the intermediate controller optimization result will in general be better than before optimization. In this way, step 3 and 4 are done much quicker, so the control engineer focuses more onto how to change e.g. the controller structure.

#### 2) LOCAL NOTCH FITTING

Experience has shown that the optimization algorithm has difficulties with using notches well. The reason for this is their local manifestation in the frequency domain. The parameters have to be set quite accurate such that the notch has the desired effect, e.g. removing a resonance peak. Due to this locality, a global optimization method has difficulties to find these parameter values. On the contrary, an experienced control engineer detects such a local problem immediately. Therefore, notch tuning is implemented as a separate optimization problem in which the user indicates the frequency range of interest. For placing a

notch, the objective is most often removing a resonance peak. With specification of the frequency interval of interest, the inverse of the notch can be fitted onto the resonance. This way, such a difficult problem from a global point of view is changed in an easy problem from a local point of view.

Within a few seconds the optimal notch is computed which again accelerates the convergence to a high performance controller.

### 3) SELECTED OPTIMIZATION

A major benefit of using optimization during loop shaping is to optimize only a selection of controller blocks of the whole controller. This will be made clear with a few examples. Suppose that all poles and zeros lie on the correct spot except that the gain is not correctly tuned yet regarding the constraints. By optimizing only the gain, the gain is tuned such that the constraints are just met.

Another possibility is to add an integrator or low pass filter to the controller and then only optimize these filters together with the gain. This way it is possible to increase the gain below the bandwidth and decrease the gain above the bandwidth without deteriorating e.g. a correctly fitted notch filter.

Another example is to add a controller block with fixed parameters and then optimize only the other controller parameters. Suppose that the control engineer knows that disturbances are present at high frequencies. By adding a low pass filter with the pole at a lower frequency, the optimization algorithm will try to achieve the best result under this controller limitation. In such a way a priori knowledge about the plant and disturbances can be incorporated in the loop shaping procedure.

## IV. APPLICATION DESCRIPTION

An application, called *Shape It*, has been written in Matlab v7 suited for optimization aided loop shaping. The GUI of *Shape It* is shown in Fig. 2. The main features of it are described below.

- Class of plants: SISO LTI models (TF, SS or ZPK systems) or FRF-data can be loaded from an example list, the workspace or a file. For SISO LTI models, the frequency range and the amount of data points can be chosen freely, whereas for FRF-data, frequency information has to be included.
- Class of controllers: Controller blocks which can be added to the controller are: gain, integrator, lead/lag, 1st order low pass, 2nd order low pass, notch and PD. Parameter values can be set per controller block. Load/Save functionality is included. Furthermore, if a notch controller block is selected it can be fitted onto a resonance peak by pressing the "fit notch" button. A figure will pop up with the magnitude plot of the plant such that the frequency interval can be selected. An additional functionality is saving the controller to a Simulink model. This model is discretized using the

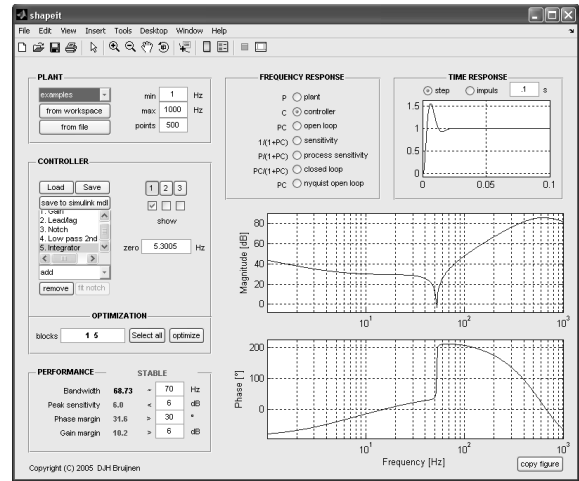


Fig. 2. GUI of Shape It

method of Tustin with prewarping properly applied per controller block. This controller can then directly be used in an experimental setup by the use of e.g. the Real-time Workshop Toolbox from Matlab/Simulink. Last but not least, three controllers can be designed simultaneously for the same plant which facilitates comparison of controllers.

- Optimization: Selection of which controller blocks are to be optimized and starting of the optimization algorithm.
- Performance indices: The performance indices are computed using both the open loop FRF and the sensitivity FRF. The computation of the bandwidth and the modulus margin are straightforward using the magnitude information of these FRFs. To compute the gain and phase margin, the phase information of the open loop function has to be determined correctly. For models, a good way to do this is to sum the phase contribution of each pole and zero of the model. Simply determining the phase of the model as a whole and then unwrapping the phase is not a robust way, e.g. the phase can shift by 360° or due to a limited frequency resolution the phase can be unwrapped incorrectly near narrow resonances. This method of determining the correct phase is used for the controller. However, the plant can consist of FRF-data for which this method can not be applied. For a correct phase margin estimation it is necessary that the phase is unwrapped correctly up to the bandwidth frequency, which requires a good FRF measurement at least up to the bandwidth. Otherwise, the objective function can not assess stability correctly, and hence, the optimization will not return a stabilizing controller.
- Visualization: To support the control engineer by interpreting the result of a controller, response figures can be generated such as Bode plots of the plant, controller, open loop, sensitivity, process sensitivity and closed loop, the Nyquist plot of the open loop and in case of using a plant model the step or impulse time response.

## V. LOOP SHAPING EXAMPLE

To test the optimization aided loop shaping tool, a non-collocated system is used as shown in Fig. 3. The system consists of two masses with a rod in between. The actuator is acting on the motor mass, while the position encoder is mounted on the load mass.

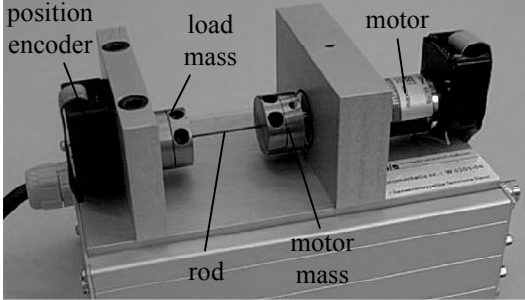


Fig. 3. Experimental setup

FRF-measurements of 3 similar non-collocated systems are shown in Fig. 4. Differences are mainly due to small variations in the mechanics of the system. Around the resonance the phase drops from  $-180^\circ$  to  $-360^\circ$  which is limiting for the attainable bandwidth.

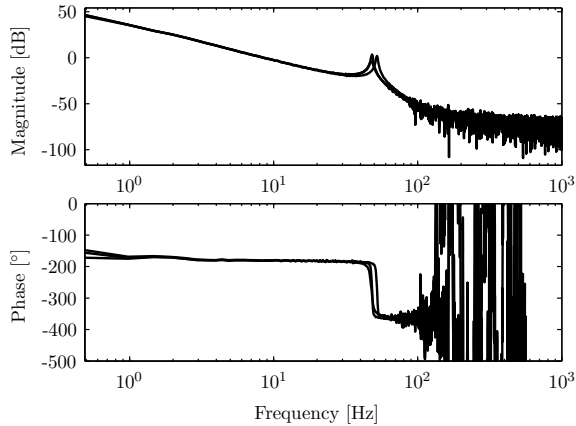


Fig. 4. Bode plot of 3 similar non-collocated systems

From experience, the control engineer knows that for this type of system a lead/lag filter is required to achieve stability. Furthermore, he knows that the resonance peak will cause stability problems when increasing the bandwidth due to phase loss. An integrator and a low pass filter are optional to increase gain below the bandwidth and decrease gain above the bandwidth respectively. The controller shown in Fig. 5 contains a gain, a lead/lag filter, an integrator, a low pass filter and a notch filter. First, the notch filter was locally fitted by optimization such that the resonance peaks are straightened as good as possible for all FRFs. Due to small differences in mass and stiffness, the resonances vary a little such that a compromising, wider notch results. After that, the other parameters are optimized with the optimization algorithm

with a target bandwidth of 10 Hz, modulus margin of 6 dB, gain margin of 6 dB and a phase margin of  $30^\circ$ .

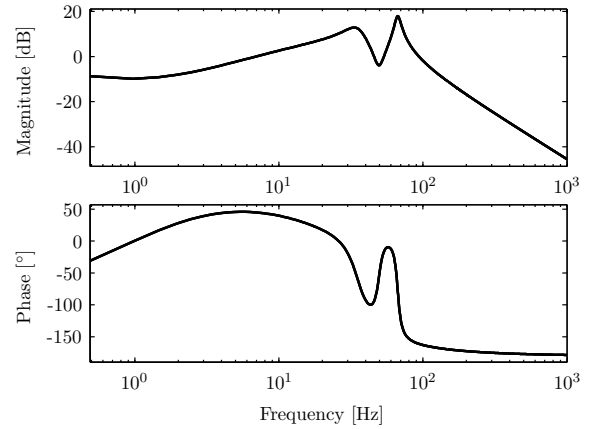


Fig. 5. Controller

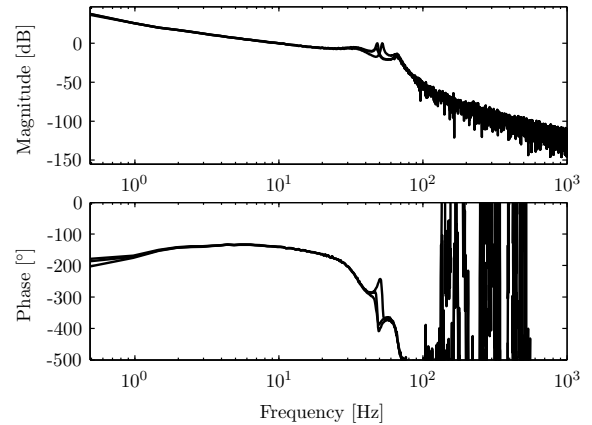


Fig. 6. Open loop

In Fig. 6 and Fig. 7 the resulting open loop and Nyquist plot are shown. The optimization algorithm managed to satisfy the target specifications for all 3 systems. The time that it takes to design this controller is only a few minutes including the optimization. The controller can be exported to ready-to-use code which makes the controller design process suited in a rapid control prototyping environment.

## VI. EXPERIMENTS

The designed controller in the previous section is exported to ready-to-use code and tested for all 3 similar setups. In Fig. 8 and Fig. 9, the experimental step response and the response to a sine of 1 Hz respectively is shown for all setups. As expected, the behavior of all setups is similar because the performance criteria of all 3 setups are similar. The step response is an indication of how fast it can react to disturbances. The rise time of the system is approximately 0.1 s which corresponds to the closed loop bandwidth of 10 Hz. Furthermore, the response to a sine disturbance shows the disturbance rejection of the feedback controller around 1 Hz. The maximum error is about 10% of the disturbance

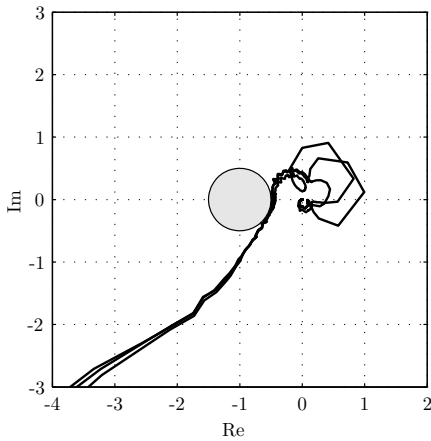


Fig. 7. Nyquist

amplitude which agrees with an open loop gain of about 25 dB at 1 Hz.

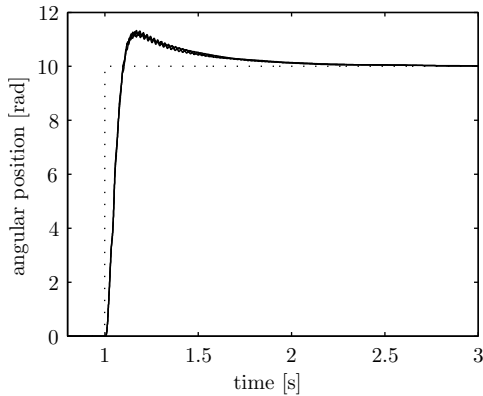


Fig. 8. Step response of the setups. The dotted line is the reference and the solid lines are the responses of the 3 setups.

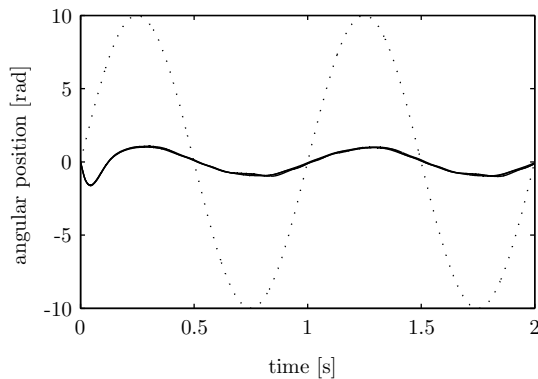


Fig. 9. 1 Hz sine disturbance response of the setups. The dotted line is the disturbance and the solid lines are the error signals of the 3 setups.

## VII. CONCLUSIONS

The optimization aided loop shaping approach as proposed in this paper and implemented in the *Shape It* tool has shown

to be a useful tool to assist the control engineer in rapidly designing a high performance feedback motion controller. The main advantages are: 1) The controller structure can be chosen arbitrarily out of a set of commonly used controller blocks, 2) measured FRF-data can be used directly for loop shaping, 3) manual loop shaping and notch placement is assisted by optimization such that the control engineer does not have to find the optimal balance between different tuning parameters which is time consuming, 4) a priori knowledge of the control engineer is easily integrated in the optimization process by fixing a selection of the controller blocks, 5) The final controller can be exported to a discretized controller using Tustin and pre-warping in Simulink ready-to-use for simulation and experimentation.

The approach can also be extended towards other manual tuning methods e.g. QFT tuning, where it can resolve one of the two major drawbacks of QFT (i.e. the loop shape part, not the grid problem). Also, combinations are possible with model-based designs, where model-based controllers could be used as starting point or reference.

## REFERENCES

- [1] D.O. Anderson and J.B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice Hall, Englewood Cliffs, NJ, USA, 1990.
- [2] H.Y. Jan C.L. Lin and N.C. Shieh. Ga-based multiobjective pid control for a linear brushless dc motor. *IEEE/ASME Transactions on Mechatronics*, 8(1):56–65, March 2003.
- [3] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-space solutions to standard  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  control problems. *IEEE Transactions on Automatic Control*, 34:831–847, 1989.
- [4] G.F. Franklin, D.J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
- [5] S. Hara, T. Iwasaki, and D. Shiokata. Robust pid control using generalized kyp synthesis: direct open-loop shaping in multiple frequency ranges. *IEEE Control Systems Magazine*, 26:80–91, 2006.
- [6] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [7] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks 1995*, pages 1942–1948, 1995.
- [8] N. Killingsworth and M. Krstić. Auto-tuning of pid controllers via extremum seeking. In *Proceedings of the 2005 American Control Conference*, pages 2251–2256, 2005.
- [9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220, 4598:671–680, 1983.
- [10] I.D. Landau, F. Rolland, C. Cyrot, and A. Voda. Digital robust control. the combined pole placement/sensitivity shaping method. French version published in *La Robustesse. Presented at the Summer Control School of Grenoble 'Robustness Analysis and Design of Robust Controllers'*, 1993.
- [11] B.J. Lurie, B.J. Lurie, and P.J. Enright. *Classical Feedback Control with MATLAB*. Marcel Dekker, Inc., New York, NY, USA, 2000.
- [12] The MathWorks. Genetic algorithm and direct search toolbox, 2003.
- [13] F.F. Syed, A.H.M.A. Rahim, and J.M. Ba-Khashwain. Robust statcom controller design using pso based automatic loop-shaping procedure. In *Proceedings of the 2005 IEEE Conference on Control Applications*, pages 440–445, 2005.
- [14] K.S. Tang, K.F. Man, S. Kwong, and Q. He. Genetic algorithms and their applications. *IEEE Signal Processing Magazine*, 13:22–37, 1996.
- [15] Q. Wang, P. Spronck, and R. Tracht. An overview of genetic algorithms applied to control engineering problems. In *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, pages 1651–1656, 2003.
- [16] O. Yaniv and M. Nagurka. Automatic loop shaping of structured controllers satisfying qft performance. *Journal of Dynamic Systems, Measurement, and Control*, 127:472–477, 2005.