Special Issue Article

Institution of MECHANICAL ENGINEERS

Journal of SYSTEMS AND CONTROL ENGINEERING

Proc IMechE Part I:
J Systems and Control Engineering
2014, Vol. 228(10) 741–757

DOI: 10.1177/0959651814527936
pii.sagepub.com
SAGE

Holistic design methodology for mechatronic systems

Daniel Malmquist, Daniel Frede and Jan Wikander

Abstract

The wide range of engineering domains aggregated in mechatronic systems can cause problems for design engineers. It is important to treat the different domains in an integrated, concurrent manner during design to be able to achieve the frequently sought-for synergetic effects of mechatronic systems. Traditional design methods are usually based on the different engineering disciplines being treated separately and only integrated at a late stage of the development process. Consequently, those methods do not work sufficiently well for mechatronic systems, leading to a suboptimal product. Previous research by the authors presents a novel approach to mechatronic system design by allowing quick optimisation and evaluation of design concepts. This is done by front loading certain design activities, hence decreasing the need for time- and cost-consuming iterations in later design stages. The method is backed up by a supporting software tool prototype. This article extends the method by including the dynamic aspects of the designed systems while also implementing basic control aspects, hence creating a concurrent and holistic method for mechatronic system design. This allows the designer to take synergetic effects into account at an earlier stage of the design process, hence increasing product quality and decreasing development costs. A conceptual design case is used in this article for an initial evaluation of the method and the results show great potential for the methodology.

Keywords

Mechatronic design, concept evaluation, integrated control, holism

Date received: 3 May 2013; accepted: 13 February 2014

Introduction

Mechatronic system design is challenging, involving multiple classical engineering disciplines with the additional need to integrate them with each other so that the system design can benefit from the synergies available through concurrent design. Current mechatronic design methodologies, for example, VDI 2206[1] and Li et al.,[2] seek to make use of this by trying to integrate the involved engineering domains as early in the product development process as possible.

General engineering design

Researchers in the field of design have developed various descriptions of the product development process. Some of the more popular models are given by Ulrich and Eppinger,[3] Pahl et al.[4] and VDI 2221.[5] Although the individual process models are different from each other, they all split the process into multiple stages, with the number of stages differing between models. Also, each model starts at an abstract level in the early development phases, with the detailed design carried out later on based on the outcome of the earlier phases. Furthermore, the mentioned development models do not regard the design process as sequential. Instead, they consider it as an iterative process where feedback from later design phases is used to improve the earlier design. As a result, whenever problems occur during later phases, the current design has to be evaluated, the problem identified and then fixed. This means that if changes to the current design are needed, an earlier design also has to be changed; in the worst case, the initial conceptual design. The design phases subsequent to the implemented change also have to be repeated in order to propagate the improved design through all design steps. This repetition increases overall product development cost. Figure 1 depicts a general design

Department of Machine Design, KTH Royal Institute of Technology, Stockholm, Sweden

Corresponding author:
Daniel Malmquist, Department of Machine Design, KTH Royal Institute of Technology, Brinellvägen 83, 10044 Stockholm, Sweden.
Email: malmqui@kth.se

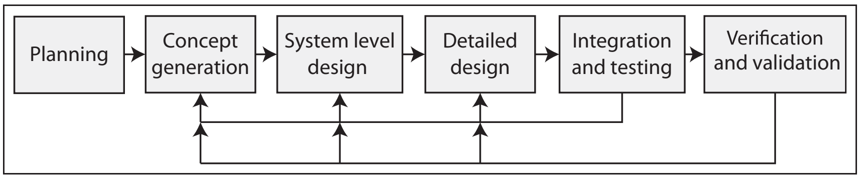

Figure 1. A generic product development process with its feedback loops to previous design phases.

process in simple terms, based on the model given in Pahl et al.[4]

A widely used model of the generic mechatronics design process is presented in the guideline VDI 2206[1] and is commonly referred to as the V-model. Adopted from software development, it is distinguished from conventional development processes in that it integrates the different domains apparent in mechatronics system design. However, similar to the classic models mentioned above, a single V-model iteration still starts by defining product requirements. As can be seen in Figure 2(a), mechatronic system design often starts in the abstract and realises the domain-specific detailed design in later phases. The V-model suggests that during the integration phases, the product's individual features are always checked, that is, verified and validated against its initial requirements. Although the VDI guideline describes an integrated mechatronic system design, it also notes that a complex mechatronic product design is generally realised by running a number of V-cycles, as shown in Figure 2(b). Therefore, the overall design process, as described by the V-model, is of an iterative nature, which reflects the findings of the more general development models.

Motivation

In order to assure its market position, every company should strive to develop innovative products. Innovation alone, however, does not guarantee profit. If development and production can be made efficient so that the corresponding cost can be kept low, chances for a successful and profitable product launch increase significantly. In order to lower design cost (and thereby overall product cost), the development time has to be shortened while still assuring the same, or even increased, product performance and quality. To achieve this goal, the mentioned iterative feedback loops to earlier phases should be avoided. In order to do this, the proposed holistic approach front loads design activities from later design stages into the early design stage. In particular, the use of a model-based multidisciplinary approach with multi-objective optimisation already included in the conceptual phase is proposed. The hypothesis is that in doing so, the number of later design changes can be reduced as a consequence of having derived a holistically optimal concept, resulting in a shorter time-to-market. Figure 3 shows that the proposed method aims at early concept evaluation and optimisation. The challenge of the early phase design lies in the fact that only a little information is known (e.g. only requirements given), and the cost associated with decisions taken here is high compared to the ones in succeeding design phases. As pointed out in Malmquist et al.,[6] there is still a lack of support for the analysis and evaluation of the various concepts generated during early design phases. Furthermore, the time needed to evaluate a concept is regarded as crucial for most engineering companies. Therefore, the proposed approach seeks to support concept evaluation in a swift manner, in particular by avoiding time-consuming simulations. This, together with the mentioned multi-objective optimisation, will lead to an improved concept so that overall development costs can be kept low. A previously published design case by Malmquist et al.,[7] based on an earlier iteration of the method, showed a potential 40% decrease in system size compared to a system designed with traditional methods.

Related research

The approach proposed in this article seeks to treat mechatronics system design in a holistically integrated manner with not only the underlying domains of mechanical and electrical engineering considered equally and concurrently throughout the design but also incorporating the important domain of control engineering. A truly integrated system design can only be obtained by considering the mutual effect of those classic mechatronic domains.

While mechanics and electronics can coexist with only limited influence on each other, this is certainly not the case when integrating control design as well. The idea of a concurrent system design including control dates back to the 1990s' research on integrated structure and control within the fields of aeronautics and robotics.

The group around Asada contributed to this concept[8] showing the necessity of modelling the dynamics of the structural system accurately and properly in order to design an appropriate controller. In Savant and Asada,[9] both the structure model's validity and the robustness margin of the proposed controller are taken into consideration for an integrated structure and control design. It was found that structural design parameters should be selected early in the design process to assure model validity of the mechanical structure, which is continuously altered during the design process. In order to be able to control the system in real time,

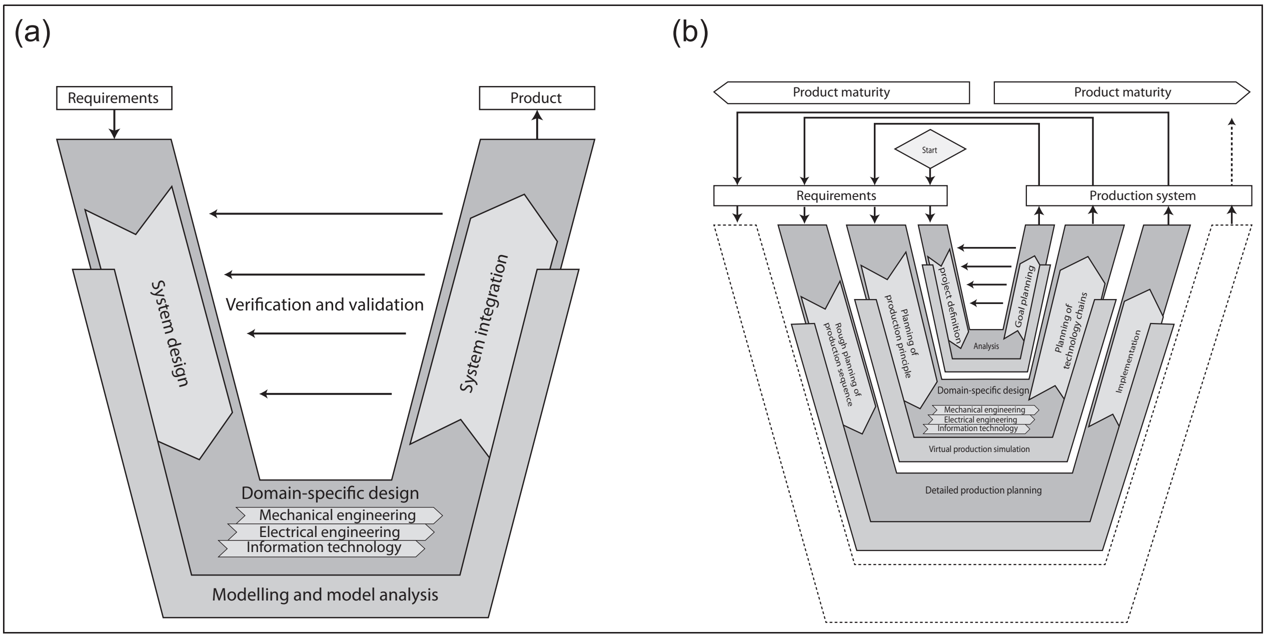

Figure 2. V-model, adapted from VDI 2206: (a) a single V-model iteration and (b) the development process considered as an iterative V-model process.[1]

the order of a proposed control design was intentionally reduced. However, such reduction needs to be handled with care, as unmodelled eigenfrequencies are often the cause of stability problems.

The work of Kajiwara and Nagamatsu[10] integrates structure and control design with consideration of performance and stability. A sensitivity analysis is used for pole placement to achieve controllability and stability. The structure is then optimised concurrently with the desired control behaviour.

Li et al.[2] present a generic model of concurrent mechatronic design, referred to as design for control (DFC), where the design process is described as a mapping of the requirement space to a structural space. The DFC methodology seeks to design the underlying mechanical structure in such a way that control design is facilitated through an appropriate dynamic model of that structure. Hence, from the dynamic performance point of view, DFC derives an optimal system, which considers both structure and control. However, the solution is determined in a time-consuming manner, iterating between structure and control design.

More recent work on the topic of concurrent mechatronic systems design by Da Silva et al.[11,12] points out that for an optimal mechatronics design, it is not only the structural and control parameters that need to be considered but also the design specification in the different domains involved.

Previous work

The starting point to the research presented here is Roos,[13] who developed a methodology for the optimal design of mechatronic servo systems when subject to given load specifications, where optimality regards various aspects such as system volume, weight and cost. However, Roos only considered one fixed system configuration and did not consider any multi-objective optimisation with the result that only one of the mentioned aspects could be used as an optimisation objective. In order to achieve an integrated design approach, the concept of model-based design is applied. As a consequence, Roos developed component models that, when lumped together, form the mechatronic system to be optimised for one of the mentioned aspects. These models relate physical properties (e.g. geometry and material) to performance properties (e.g. torque transfer capacity). Moreover, algebraic models are used so that the resulting equation systems can be solved without applying higher calculus, avoiding time-consuming solving of differential equations. In order to obtain a realisable set of solutions, Roos made use of scalable models, calculating (extrapolating) the physical properties of a motor of an arbitrary size from a real motor's properties. In terms of dimensioning the individual components, Roos applied standard mechanical engineering guidelines (e.g. peak torque constrains gear design and root mean square (RMS) torque rules motor design). Since the approach presented in Roos[13] is only capable of handling the fixed configuration of mechatronic servo systems, wider application within mechatronic system design is limited, especially for evaluation of various differently configured mechatronic system concepts. However, the models for direct current (DC) motors and for the design of planetary gears used in this research are based on Roos.

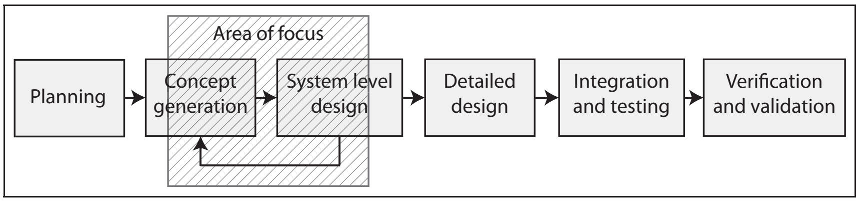

Figure 3. Area of focus to which the proposed method applies.

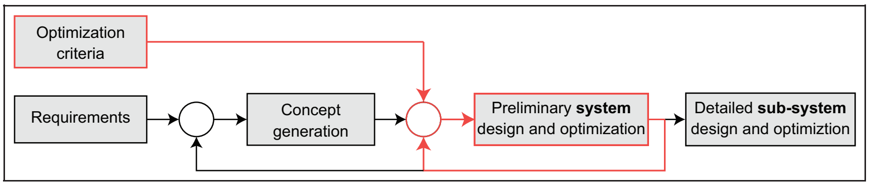

Figure 4. A simplified outline of the proposed method. The generated concepts are holistically designed and optimised.

Malmquist et al.[6] extend Roos' work, finding that support for concept evaluation is in demand by various engineering companies. In order to be helpful during earlier phases, a support tool should be simple to use and also quick in its internal calculations. Therefore, as a concrete target, analysing and evaluating concepts within minutes is focused on. Making use of linear models was deemed inevitable to fulfil this requirement. Malmquist et al.[6] developed a software framework prototype that extends the work of Roos making arbitrary system configurations possible to analyse and optimise for multiple objectives.

Frede et al.[14] developed these thoughts further, presenting a concept on how to also include dynamic behaviour on top of the static dimensioning of a mechatronic system. While in the static case, peak and quadratic mean values of the load torque ruled the resulting mechatronic system, now the frequency spectrum of the load is taken into account as well. Moreover, the crucial aspect of integrating a controller by closing the loop was discussed. It should be highlighted that in contrast to other research on integrated structure and control, Frede et al.[14] regard the controller as a specification to the overall system rather than an optimisation objective.

The work presented in this article further extends the previous work about integrating control aspects, and the results from implementing this into the software framework are presented. This is a major step towards a holistic mechatronics design approach.

Proposed methodology approach

Context of early system design

As explained earlier, the proposed approach aims to evaluate early mechatronic concepts at a reasonable cost (i.e. time). The evaluation regards the concept's potential to fulfil the system requirements and specification as a final product. However, the method does not include idea and subsequent concept generation; rather it assumes a concept, or a set of concepts, to be given.

In order to shorten the time-to-market of a product, its development process should be as efficient as possible. As such, the early concept is not only evaluated and analysed for fulfilment of its requirements but also should be optimised for some of the same, so as to avoid iteration loops during later design phases. The requirements used in the presented method relate both to static properties (e.g. a nominal motion profile which the concept needs to be able to follow) and dynamic behaviour (e.g. the desired closed-loop poles are given). Figure 4 depicts how a preliminary system design is derived by optimising the selected concepts. Once an optimal system is calculated, according to the general design procedure, the optimal conceptual design is generated in detail, using more accurate models, applying simulation and detailed component analysis. The detailed design is out of scope of this article.

Modelling approach

In order to meet the time constraints on concept evaluation, the proposed method avoids time-consuming differential equation-solving for each point in time: the concept is evaluated without simulating it. Therefore, both models and computational methods underlying the evaluation need to have limited complexity. According to Janschek,[15] the system to be designed can be represented by linear models when treated in early design phases without losing too much detail. Furthermore, for many mechatronic systems, those simplifications using linearised models can be implemented by means of multiple mass oscillator chains, as depicted in Figure 5. However, it is important to note that most, if not all, mechatronics systems are far more complex, and thus follow a non-linear behaviour, which is not covered with this simplified representation. Still, the dominant behaviour and critical points of

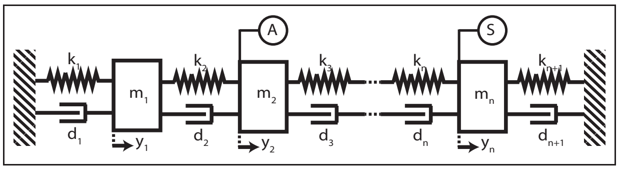

Figure 5. A translational multiple mass oscillator that is used to model early concepts.

operation can be well identified, which is considered as sufficient for early analysis.

As the methodology assumes the concept idea(s) to be given as an outcome of a prior design phase, the design engineer only needs to specify requirements and design goals as well as configure the component system representing the concept. The latter is done by picking the desired mechatronic base components from a library and configuring how they are interconnected. In addition, the component parameters and optimisation variables and objectives need to be specified. Currently, the component library spans the classic mechatronic domains of mechanical engineering, electrical engineering and control. Base components found in the library are, for instance, electrical actuators, mechanical transmissions, structural elements, sensors (currently limited to sensor location only) and controllers. Figure 6 illustrates the idea of having a component library from which to model the given concept idea.

It is important to note that every base component model internally implements two different model concepts. In order to capture static properties, the so-called algebraic component property models were developed. These cover physical properties related to component performance such as weight, volume or material cost. The component behaviour is modelled by dynamic component models. For these, the linear model approach for conceptual modelling is applied. In the mechanical domain, these can be described as multiple mass oscillators. For other mechatronic domains, this concept is adapted and network theory analogies are applied, as outlined in Lenk et al.[16] When combining different basic mechatronic component models to model the desired concept, the overall behaviour is expected to be more complex. However, the resulting system can still be described by linear differential equations, though now of a higher order. In Figure 6, the modelling idea including the two modelling paradigms immanent to every component is also illustrated.

One of the requirements the mechatronic concept should be designed for is the nominal motion profile. Following the idea of efficient evaluation and analysis, this motion profile needs to be simplified. Here, however, the simplification does not lie in linearisation. Instead, the profile is treated as periodic load, which the mechatronic system is subject to. As will be shown in the next section, when integrating control design into the evaluation/analysis, it is advantageous to analyse the load profile for its dominant frequencies. Thus, a Fourier analysis is applied on the motion profile, resulting in a set of sinusoidal signals, defined by their frequency, amplitude and phase shift. According to the theory of Fourier analysis, superimposing these signals will give an approximation of the original signal, where the accuracy depends on the number of harmonics superimposed to reflect the original signal.

Control approach

The methodology is novel in the sense that it treats mechanical, electrical and control domains concurrently. Previous work on integrated structure and control, as outlined in section 'Related research', focuses on control performance while also having structural parameters as free variables during the control design process. However, the optimisation of aspects subject to both structural and behavioural parameters has only been explored to a rather limited extent. By integrating control into system design, a holistic mechatronic system design is now achieved, applicable to early design concepts. To facilitate time-efficient system-level concept analysis and optimisation, the controller design is considered as a specification. Hence, optimality in design targets overall system criteria, which the control system contributes to with given specifications. Of course, the optimality of the solution is limited by the accuracy of the simplified linear models used for describing the overall concept.

The dynamics of the system are evaluated by selecting a controller structure, determining the closed-loop system transfer functions and then using the information from the Fourier approximation to determine the

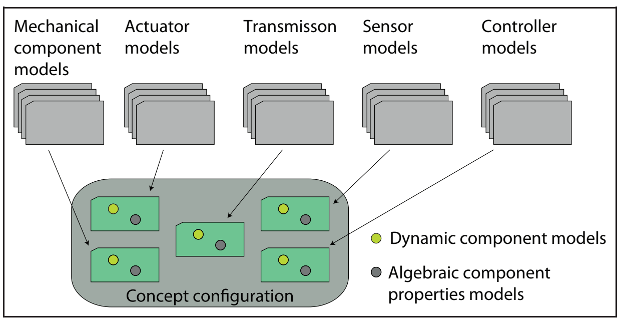

Figure 6. Concept is built by picking and configuring models from a given model library.

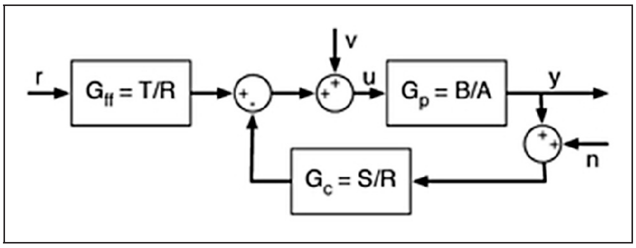

Figure 7. Two-degree-of-freedom controller structure as used in the proposed approach.

output of the transfer function without simulation. The response of a linear system to a set of overlaid (concurrent) sinusoidal inputs is formed by superimposing the responses of each individual input. Hence, the transfer function's output can be constructed. Different transfer functions can be used to evaluate different dynamic characteristics, such as control error, disturbance/noise rejection and actuator input saturation.

The design of controller structure and controller parameters needs to be known in order to determine the closed-loop transfer functions. Several different approaches for this should work with the methodology; the one described here is the one tested by the authors so far. A fixed, 2-degree-of-freedom control structure is used while the controller parameters are determined by polynomial pole placement. The pole locations are considered a part of the system specification for now. See Figure 7 for the closed-loop system structure where G_p is the open-loop system transfer function, G_{ff} is the feedforward transfer function and G_c is the feedback controller transfer function.

The pole placement method used is adapted from Åström and Wittenmark[17] and follows the following steps:

1. Select control structure

$$G_c(s) = \frac{S(s)}{R(s)} = \frac{S_n s^n + \cdots + S_0}{s^n + r_{n-1}s^n + \cdots + r_0} \tag{1}$$

where n is the number of poles in G_p, and S_i and r_i are controller parameters.

2. Calculate the closed-loop polynomial

$$A_{cl} = A \cdot R + B \cdot S \tag{2}$$

where A and B are denominator and numerator of the process transfer functions, respectively (see Figure 7).

3. Select a desired closed-loop polynomial

$$A_d = A_m(s)A_o(s) \tag{3}$$

where A_m is a polynomial of order equal to that of A, and A_o is a polynomial of order equal to the order of A_{cl} minus the order of A_m.

4. Solve the Diophantine equation $A_{cl}(s) = A_d(s)$ for the controller parameters in S and R.
5. Set the feedforward polynomial to

$$T(s) = t_0 A_o(s) \tag{4}$$

where t_0 is a static gain that gives unit DC gain in the closed-loop transfer function from reference r to output, y, for example

$$t_0 = \frac{A_m(0)}{B(0)} \tag{5}$$

This pole placement method is carried out analytically so that it does not need to be computed every time a pole location or system parameter value changes. This allows most of the calculations involved to take part outside the actual optimisation loop, hence increasing the speed of it.

Optimal system-level design

The above-introduced ideas on time-efficient concept evaluation, physical component modelling and control design are combined with optimisation to derive a method for optimal system-level design suitable for evaluation of concepts. The overall process, as depicted in Figure 8, is as follows. The concept idea is modelled by means of connecting and configuring the components provided in a common library. System requirements need to be specified to describe not only the motion profile the system is designed for but also the specification of the dynamic system behaviour. In addition, the optimisation objectives are defined. If desired, the solver optimises multiple objectives concurrently. As shown in Figure 8, the user is only required to manually specify the concept idea given the component library and the system requirements.

The optimisation is based on the theory of genetic algorithms[18] and is designed to be time-efficient as well as flexible with regard to the inner component model description. The objective function is evaluated and solved in such a way that first a static dimensioning is carried out by means of the algebraic component property models. Its results are then used to check whether the derived system would also fulfil the requirements of dynamic behaviour. For that purpose, a dimensioning factor is introduced which over- or under-dimensions the optimal static system to also fulfil the dynamic specifications. From all possible solutions fulfilling both static and dynamic requirements, the optimiser is supposed to pick the best solution – with regard to the

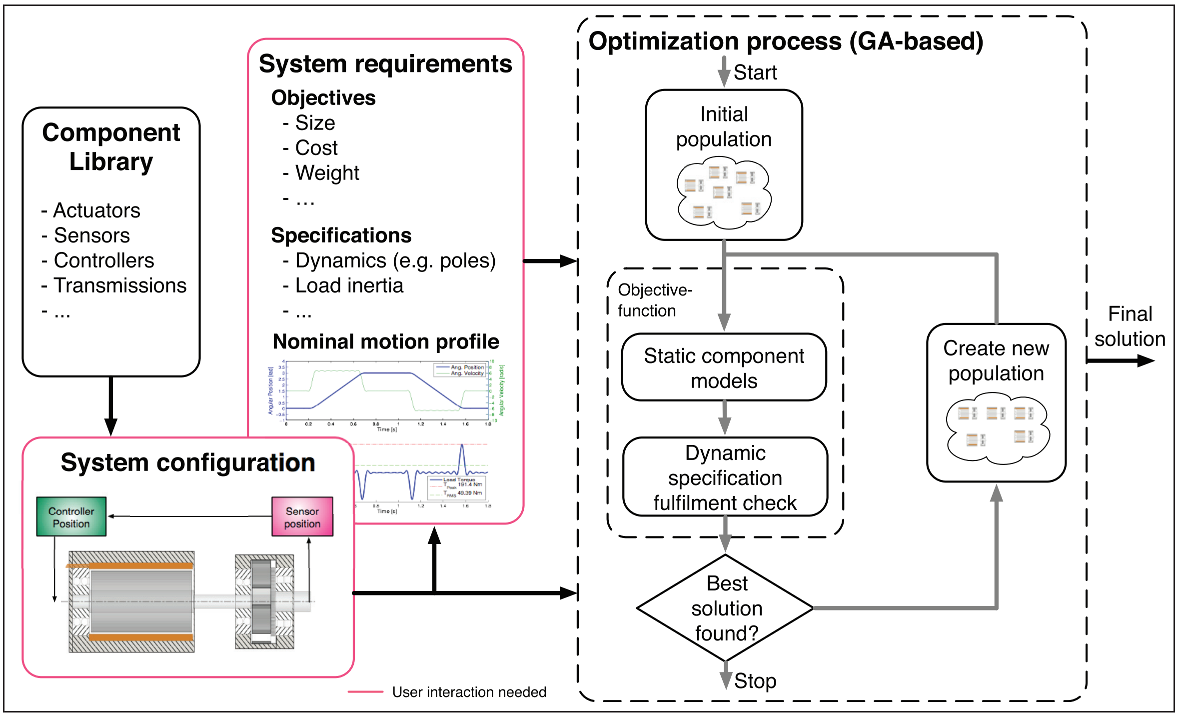

Figure 8. Overall outline of the proposed methodology.
GA: genetic algorithm.

given optimisation objective – and present it to the user as a final solution.

Framework

The methodology described is supported by a prototype tool framework, which has been developed by the authors to validate the approach. An earlier iteration of the tool is described in Malmquist et al.[6] and Frede et al.,[14] and therefore, only a short overview will be given in this article in addition to details about improvements made in the newest version. The tool has been implemented as a comprehensive MATLAB toolset where each component has its static models defined as MATLAB functions, and dynamic models defined as Laplace-transformed equations written in plain text. Each component has its respective characteristics and different models defined in its own XML file, which also describes what inputs and outputs are needed or given by each model. The system configuration, optimisation objectives, design variables and dynamic specifications are all given in another XML file. The XML format allows for simple and quick implementation of new configuration features as well as simplifying implementation of new tools using the same format. Languages such as SysML and Modelica were considered, but deemed unnecessarily complex for this prototype, especially since they were originally designed for other purposes.

The prototype tool design framework facilitates the following:

- Given algebraic models of the physical components' properties and dynamic models of the components' behaviour, the hardware part of a mechatronic system is configured. These models facilitate load-based static dimensioning of the individual components.
- Given algebraic models of the physical components' properties and dynamic models of the components' behaviour, the control part of a mechatronic system is configured. These models enable the tool to validate that the hardware, designed with the static models, satisfies the dynamic specifications. In addition, they also facilitate determining a controller structure and analysing control performance against specifications.
- Given hardware and control models whose dependencies are clear, holistic system optimisation can be performed. This is carried out by combining models of the above-mentioned types and optimising for given design variables and load specification.

The main improvement from previous iterations of the tool is the ability to quickly determine dynamic transfer functions for key variables in the system, such as from sensor noise to output position or reference to actuator input. This is done by combining the dynamic equations of all the system components into one large equation system and analytically solving for the desired variable(s). The solution is calculated before the actual optimisation loop is started and fed forward to the objective function used by the optimiser. This allows

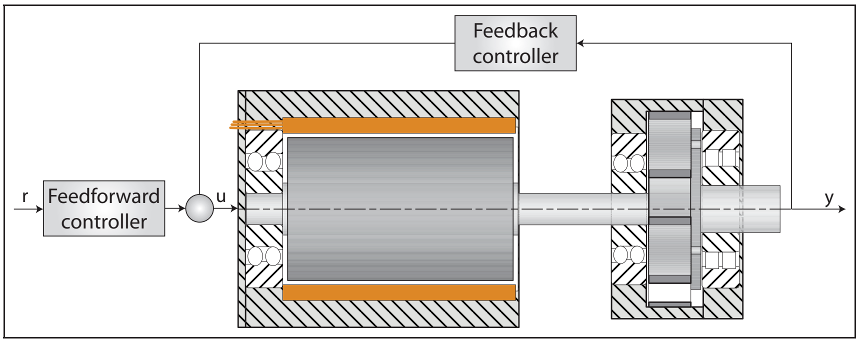

Figure 9. Mechatronic servo system with controller, which serves as the design case.

the optimiser to check in a very fast manner whether the static hardware solutions fulfil to the dynamic specifications given by the user. The ability to generate different transfer functions for the system allows characteristics such as disturbance rejection, actuator saturation and control error to be analysed.

Design case

Problem definition

A servo system consisting of a motor, a shaft and a planetary gear, as shown in Figure 9, is used to demonstrate the capability of the methodology and supporting tool framework. Component models for electric motor amplifiers have previously been developed by Roos,[13] but are excluded from this case. The optimised design variables are gear ratio, gear radius, shaft radius and dimensioning factor (described earlier). The load needs to be specified as a motion profile and a load inertia. The approximated load position profile used can be seen in Figure 10 together with the calculated load torque based on a load inertia of 1.1 kg m^2. The dynamics are specified with desired pole locations. The goal of the optimisation is to minimise the system size, that is, volume. The dynamics were evaluated with transfer functions from reference to actuator input and position output. It should also be possible to analyse, for example, disturbance rejection in a similar fashion, although this has not yet been implemented.

Models

As stated earlier, two different kinds of models are used for the methodology: static models that describe the relationship between properties such as load handling capability and size, and dynamic models describing the behaviour of the components over time. Each of the components used needs to have both types of models defined. The static dimensioning models of the electric motor (cf. section 'Motor models') and the planetary gear (cf. section 'Planetary gear models') used in this example are taken from Roos,[13] where a more detailed explanation and derivation of the models which are considered given in the context of this work can be found. Although the type of static models used for the methodology could describe many different kinds of properties (e.g. cost, weight and life expectancy), the ones used in this design example all relate to component volume. Note that the following models use the index *in* and *out* for denoting a component's input and output, where output is the port on the load side. Furthermore, the RMS of a value is denoted by a ~, while the peak value is denoted by ^.

Motor models. The static motor model, as derived in Roos,[13] gives the relationship between size and permissible RMS torque. The dominating dimensioning constraint for the motor is that the rated torque, $T_{m,rated}$, needs to be at least equal to the expected RMS torque $\tilde{T}_m$

$$T_{m,rated} \geqslant \tilde{T}_m \tag{6}$$

Roos et al.[19] derive the motor's rated torque based on mechanical, magnetic and thermal effects as

$$T_{m,rated} = C_m l_m r_m^{2.5} \tag{7}$$

where C_m is a motor type–specific constant for the same cooling conditions, l_m is the motor's rotor length and r_m is the radius of the stator. In terms of thermal considerations, equation (7) only takes resistive losses and radial heat transfer into account hence neglecting, for example, magnetic and mechanical losses as well as longitudinal heat transfer. The motor's RMS torque is derived from the torque balance as

$$\tilde{T}_m = \sqrt{\frac{1}{\tau}\int_0^\tau \left((C_{mj} l_m r_m^4 + J_m)\ddot{\varphi}_{m,out} + T_{m,out}\right)^2 dt} \tag{8}$$

where C_{mj} is a constant for a specific machine type and derived from a reference motor of the same type and under the same cooling conditions, τ is the cycle time of the load profile, $\ddot{\varphi}_{m,out}$ is the angular acceleration of the motor output shaft, $T_{m,out}$ is the load torque the motor is subject to and J_m represents all the motor inertia.

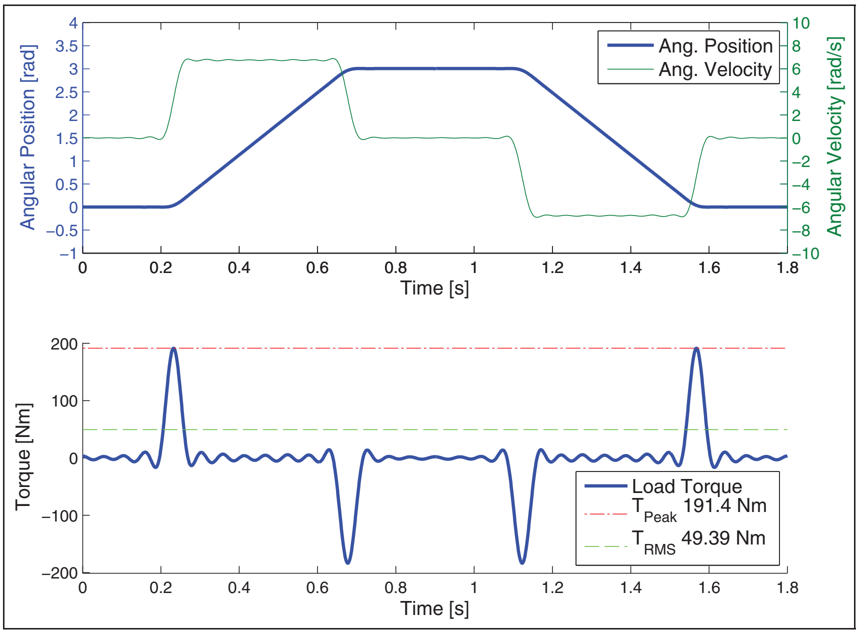

Figure 10. Load profile used for the design case.

Combining equations (6)–(8) results in

$$C_m l_m r_m^{2.5} \geqslant \sqrt{\frac{1}{\tau}\int_0^{\tau}\left((C_{mj} l_m r_m^4 + J_m)\ddot{\varphi}_{m,out} + T_{m,out}\right)^2 dt} \tag{9}$$

Equation (9) is solved for the minimum l_m, resulting in a function, which is used in conjunction with a form factor constraint such as

$$0.5 \leqslant \frac{l_m}{r_m} \leqslant 5 \tag{10}$$

to limit the set of possible solutions. With regard to system optimisation, any geometry-related parameter of the model can be considered as an optimisation variable.

The dynamics are simplistically modelled as

$$K_t i = T_{m,out} + J_m \ddot{\varphi}_{m,out} \tag{11}$$

where K_t is the motor constant and i is the electric current.

Planetary gear models. In addition to the electric motor model, Roos[13] also derives models for spur and planetary gears. This design example only uses the planetary gear model since their volume, weight and inertia are less compared to spur gear pairs for transmitting the same torque as well as enabling higher gear ratios for the same size. In contrast to the motor, the planetary gear is dimensioned based on maximum permissible peak torque.

The volume of the gearbox is approximated as

$$V_g = \pi r_g^2 b \tag{12}$$

where r_g is the outer gear radius and b is the total width of the gearing. The design guidelines for spur gears SS 1863[20] and SS 1871[21] conclude that the gear size is limited by mechanical fatigue, the bending stress in the root of a gear tooth as well as the Hertzian pressure at the teeth contact surfaces. However, Roos[13] notes that if the number of sun gear teeth is small and/or the gears are made from a somewhat ductile steel, the Hertzian pressure will likely be the constraining factor for the gear design. Roos[13] gives the following condition, which needs to be fulfilled for every teeth surface in contact

$$r_g^2 b \geqslant Z_H^2 Z_M^2 Z_\varepsilon K_{H\alpha} K_{H\beta} \frac{\hat{T}_{g,out}(n-1)^2}{6(n-2)\sigma_{H,max}^2} \tag{13}$$

where Z_H is a form factor for Hertzian pressure; Z_M is a material factor for Hertzian pressure; Z_ε is the contact ratio for Hertzian pressure; $K_{H\alpha}$ is a factor describing the division of load between teeth; $K_{H\beta}$ is a load distribution factor for Hertzian pressure; $\sigma_{H,max}$ is the maximum allowed flank pressure, which can be determined for specific gear materials; n is the gear ratio and T_g is the torque the gear is subject to. Equation (13), together with a form factor constraint that relates gear radius r_g to total gear width b, is then sufficient to calculate the volume of a specific gearbox. As with the motor model, any parameter regarding geometry can be considered as an optimisation variable for the overall system design.

The dynamic model used in this design case only takes inertia and gear ratio into account, neglecting other effects such as friction and stiffness. The model used is

$$(T_{g,in} - J_g \ddot{\varphi}_{g,in})\eta n = T_{g,out} \tag{14}$$

$$\varphi_{g,in} = n \varphi_{g,out} \tag{15}$$

where η is the gear efficiency.

Shaft models. The models used for the static dimensioning of the shaft are derived from classic solid mechanics. The following equations are taken from Collins[22]

$$r_s = \sqrt[3]{\frac{2\hat{T}_{s,out}}{\tau_{s,max}\pi}} \tag{16}$$

$$J_s = \frac{\pi r_s^4}{2} \tag{17}$$

$$G = \frac{E}{2(1+\nu)} \tag{18}$$

$$k = \frac{GJ_s}{l} \tag{19}$$

where r_s is the shaft radius, $\hat{T}_s$ is the maximum transferred torque, $\tau_{s,max}$ is the maximum permissible shear stress, J_s is the shaft inertia, G is the shear modulus, E is Young's modulus, ν is Poisson's ratio, l is the length of the shaft and k is the stiffness of the shaft. A fixed value is used for the shaft damping, d. Note that these equations describe static properties, which could be used both as an optimisation objective and as parameters in the dynamic models.

The shaft's dynamics are approximated as a mass–spring–damper modelled according to Figure 11. The resulting dynamic equations are given as

$$\frac{1}{2}J_s\ddot{\varphi}_{s,in} = T_{s,in} - k(\varphi_{s,in} - \varphi_{s,out}) - d(\dot{\varphi}_{s,in} - \dot{\varphi}_{s,out}) \tag{20}$$

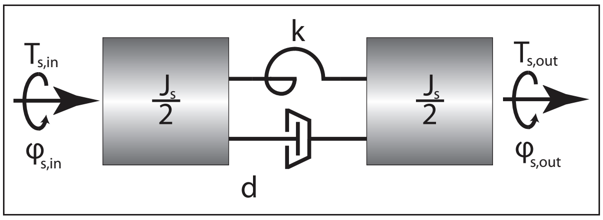

Figure 11. Lumped parameter shaft model for modelling dynamics.

$$\frac{1}{2}J_s\ddot{\varphi}_{s,out} = k(\varphi_{s,in} - \varphi_{s,out}) + d(\dot{\varphi}_{s,in} - \dot{\varphi}_{s,out}) - T_{s,out} \tag{21}$$

Load model. Since the load is not a component in the same sense as the above-mentioned ones, it does not have static models. However, a dynamic model describing the load inertia behaviour is defined as

$$J_l\ddot{\varphi}_{l,in} = T_{l,in} - T_{l,out} \tag{22}$$

$$\varphi_{l,in} = \varphi_{l,out} \tag{23}$$

where J_l is the load inertia, φ_l is the position and T_l is the load torque.

Optimisation results

Static property–based optimisation. The system was first optimised by only taking the static properties into account, neglecting the dynamics, to allow comparison of results from optimisations with and without control. The total system volume was used as the minimisation objective while gear ratio and gear radius were used as design variables for this optimisation. Figure 12 shows

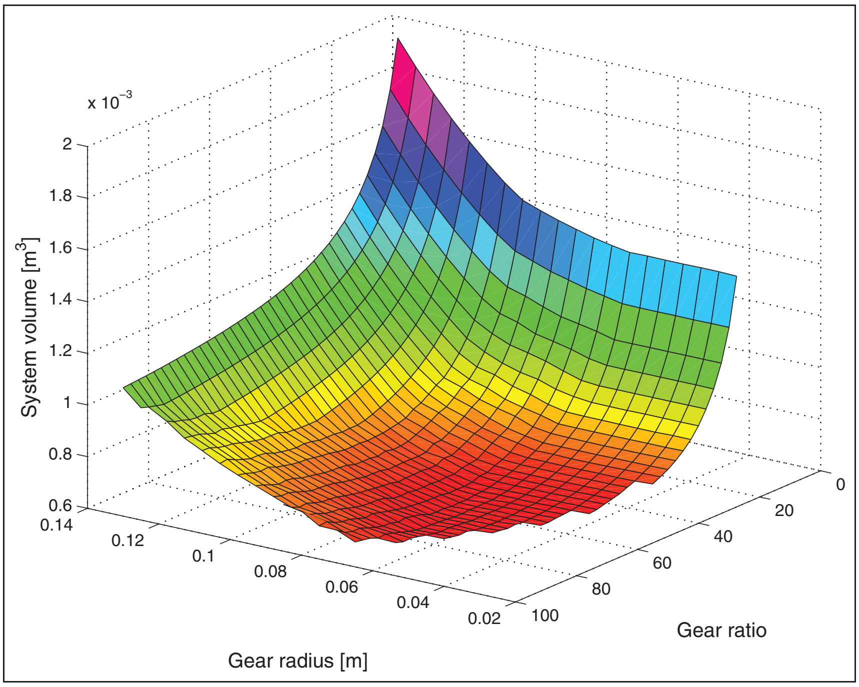

Figure 12. Result plot for case of static property optimisation.

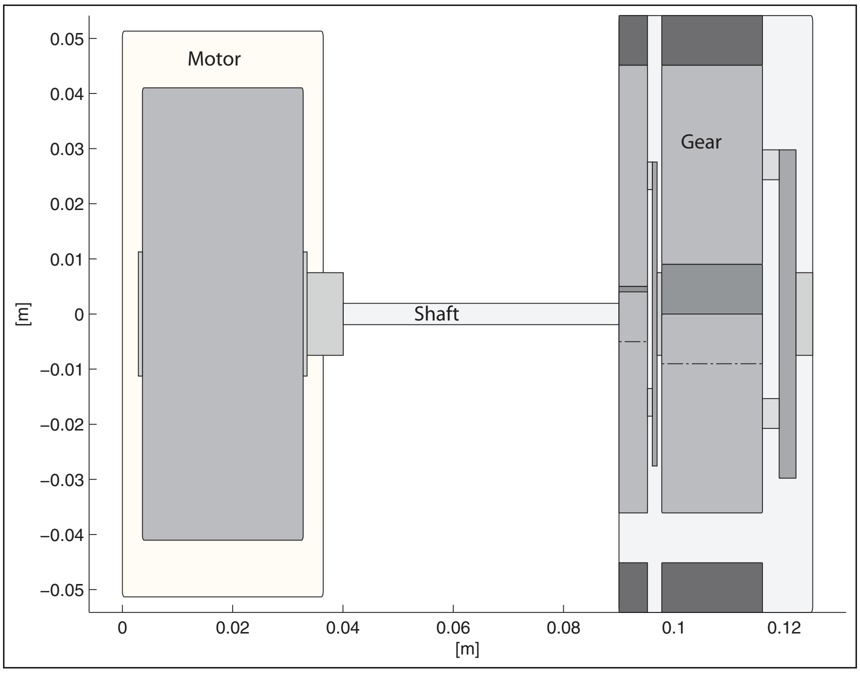

Figure 13. Concept visualisation that represents the found solution for static property optimisation.

a three-dimensional (3D) plot of system volume versus gear ratio and gear radius with the best gear radius selected for each data point. There is a minimum around gear ratio $n = 60$ and gear radius around $r_g = 0.05\,\mathrm{m}$.

Figure 13 shows the resulting system from the optimisation. The design variable values that the optimiser found were gear ratio $n = 60.61$ and gear radius $r_g = 0.0541\,\mathrm{m}$. This correlates well with what can be seen in Figure 12. The resulting shaft radius for this solution was around $r_s = 0.002\,\mathrm{m}$. The shaft radius is determined from the minimum radius required for transferring the load according to solid mechanics (since dynamics are omitted in this optimisation run). Hence, it is unrealistic to use this value in a real design case due to, for instance, its low stiffness.

An optimisation run with this specific set-up, that is, no dynamics and two design variables, consistently takes around 35 s to run on a mid-range PC at the time this article was written.

(25)–(27), where G_{yr} is the closed-loop system transfer function from reference to output position; G_{ur} is the closed-loop system transfer function from reference to actuator input and p_1, p_2, p_3 and p_4 are the desired pole places. The transfer function from reference to output position, that is, equation (26), is used to determine the integrated square error (ISE), and the transfer function from reference to actuator input, that is, equation (27), is used to determine that the dynamic load is not larger than the static dimensioning allows for, that is, that the dimensioning factor is high enough. ISE is a measure of control performance and is defined as

$$ISE = \int_0^{\tau} (r(t) - y(t))^2 dt \tag{24}$$

where $r(t)$ is the controller reference signal and $y(t)$ is the system output. As can be expected, and as shown in Figure 20, the ISE shrinks with faster poles

$$G_p = \frac{4k_t n\eta(k + ds)}{s^2\left(4k\left(J_l + n^2\eta(J_m + J_g + J_s)\right) + 4d\left(J_l + n^2\eta(J_m + J_g + J_s)\right)s + (2J_m + J_s)\left(2J_l + n^2\eta(2J_g + J_s)\right)s^2\right)} \tag{25}$$

$$G_{yr} = \frac{p_1 p_2 p_3 p_4 (k + ds)}{k(p_1 - s)(p_2 - s)(p_3 - s)(p_4 - s)} \tag{26}$$

$$G_{ur} = \frac{p_1 p_2 p_3 p_4 s^2\left(4k\left(J_l + n^2\eta(J_m + J_g + J_s)\right) + 4d\left(J_l + n^2\eta(J_m + J_g + J_s)\right)s + (2J_m + J_s)\left(2J_l + n^2\eta(2J_g + J_s)\right)s^2\right)}{k_t n\eta k(p_1 - s)(p_2 - s)(p_3 - s)(p_4 - s)} \tag{27}$$

Dynamic behaviour and static property–based optimisation. The resulting fourth-order transfer function models generated by the dynamic solver from the components' dynamic models are given in equations

As in the static case, the total system volume was used as the minimisation objective for the optimisation. However, now the shaft radius is also used as a design variable in addition to gear ratio and gear radius.

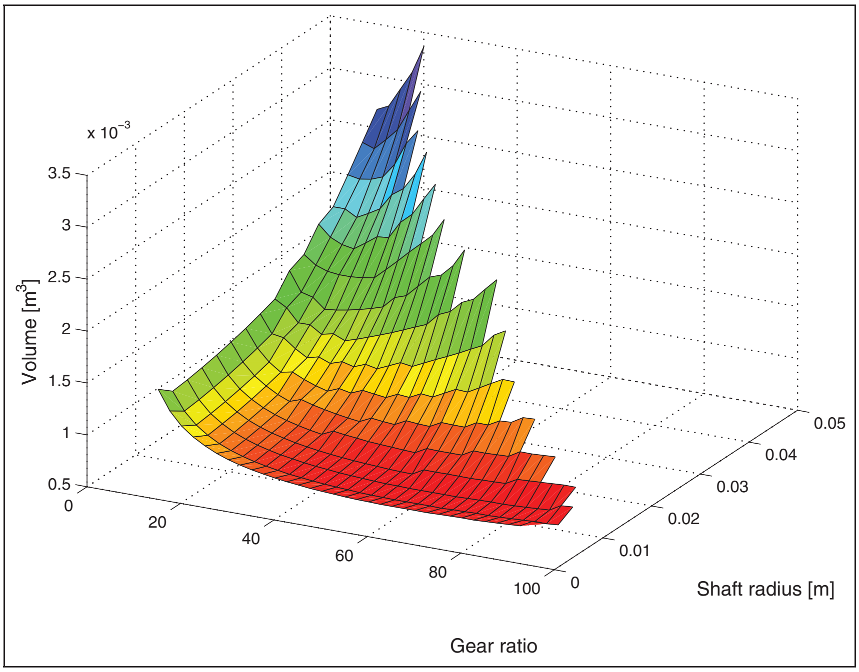

Figure 14. 3D plot for the resulting system volume. There is a minimum at gear ratio $n = 60$ and shaft radius $r_s = 0.05$ m.

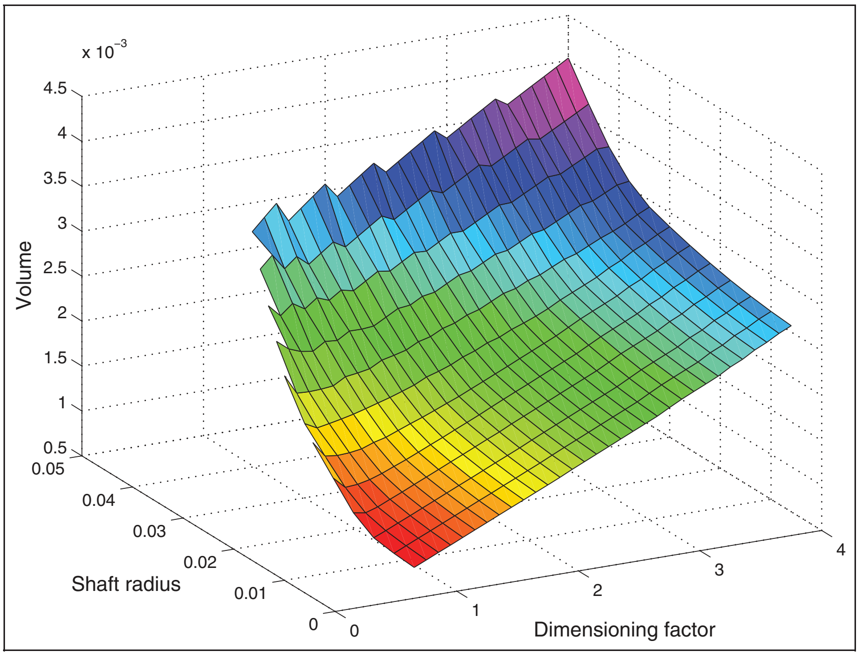

Figure 15. 3D plot for the resulting system volume as a function of shaft radius and dimensioning factor. The minimum is found for both small shaft radius and small dimensioning factor.

Before running the optimiser, the objective function was tested by plotting its results for a range of values for gear ratio, shaft radius and dimensioning factor, while keeping the gear radius fixed. The pole locations can be chosen independently from each other, but for this test they were all placed at $p_i = -150$. The results are highly dependent on pole location, and this is therefore only one of many possible results. Figure 14 shows a 3D plot of volume versus gear ratio and shaft radius. Since only two design variables can be visualised in a 3D plot, the best dimensioning factor value for each data point was used and the gear radius was fixed. Similarly, Figure 15 shows volume versus shaft radius and dimensioning factor while the best gear ratio is used. In Figure 14, there is a minimum at around gear ratio $n = 60$ and shaft radius around $r_s = 0.01$ m. The shaft radius is now determined by not only the torque it needs to transfer (like in the static case) but also the

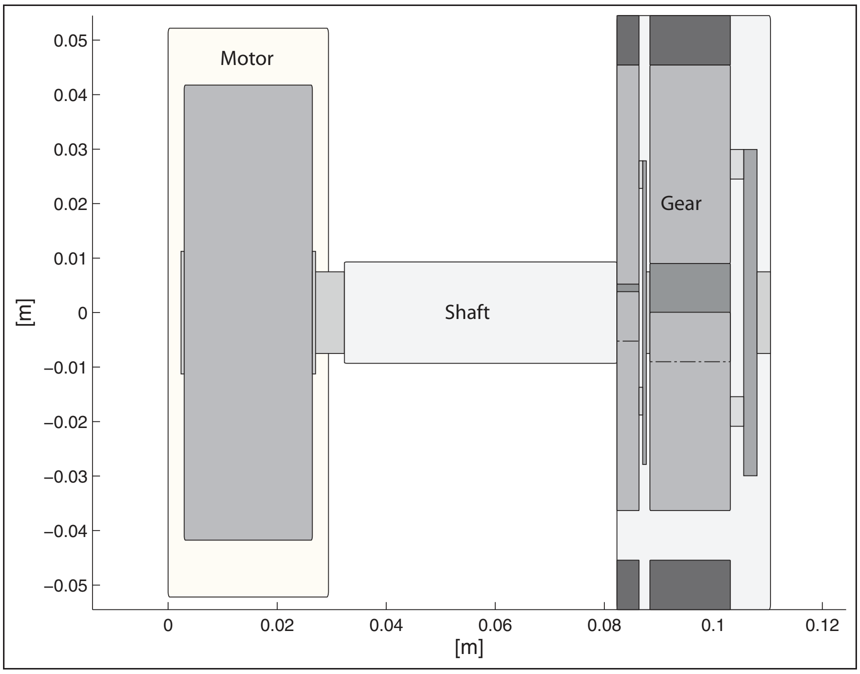

Figure 16. Concept visualisation, which represents the found solution for dynamic behaviour and static property optimisation.

dynamic constraints, the ISE in this case. A smaller shaft radius gives a more flexible shaft (i.e. lower shaft stiffness), which results in higher requirements on controller and actuator. This explains why the shaft radius is now larger than in the static case. Figure 14 also shows that a combination of high gear ratio and high shaft radius will not be able to fulfil the dynamic requirements, which is most likely the result of the shaft inertia becoming very large in comparison to the load inertia for high gear ratios as acting on the motor side since the load inertia will be seen as multiplied by $1/n^2$ due to the gearing.

Figure 16 shows the system resulting from the optimisation. The design variables given by the optimiser are gear ratio $n = 59.01$, gear radius $r_g = 0.0545\,\text{m}$, shaft radius $r_s = 0.0093\,\text{m}$ and static dimensioning factor 0.815, which correlates well with what is shown in the figures above.

Figures 17 and 18 show the output from G_{yr} and G_{ur}. It can be seen in Figure 17 that the closed-loop system lags behind to some extent for the given poles. However, as can be understood from equation (26), this is mainly due to the selected pole locations. The fact that the system is not as fast as the reference signal also explains why a dimensioning factor lesser than 1 is optimal. Figure 18 shows that actuator input is slightly lesser than what the static dimensioning used. However, this can be explained by the somewhat lower acceleration of the closed-loop system together with the calculated peak and RMS values of torque in the static and dynamic case, as shown in Table 1. The dynamic RMS is very close to the static case RMS, which therefore becomes the limiting factor.

A number of optimisations with different pole locations specified were run to investigate how volume varies with dynamic specification. Figure 19 shows the resulting component sizes for a number of different pole locations. For simplicity, the four poles were all placed in the same location for each run. As expected, faster poles put higher demands on actuator effort, and hence, the system needs to be dimensioned for a higher load, that is, higher static dimensioning factor.

Figure 20 shows the ISE of the closed-loop system.

An optimisation run with this specific set-up takes consistently around 80 s to run on a mid-range PC at the time this article was written. However, the current implementation should be regarded as an early prototype and is thus far from as computationally efficient as it has the potential to be.

Conclusion and discussion

Analysis of results

The results from the design case clearly show that the static modelling and optimisation are useful and efficient but at the same time insufficient for dimensioning systems under closed-loop control. The design case also clearly shows the strong coupling between physical system design and control system design, and that optimisation of such systems must treat both subsystems concurrently.

In the design case, the system resulting from the static and dynamic optimisation had a much larger shaft radius than the system resulting from the optimisation using only static dimensioning. Depending on the dynamic specification, it is in many cases possible for some system components to be smaller than the static dimensioning comes up with, since the

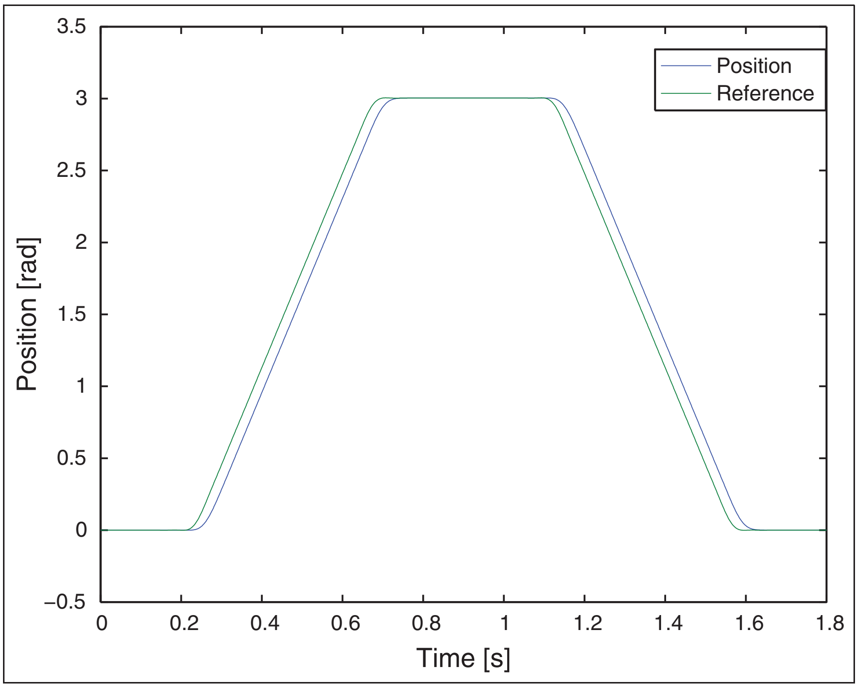

Figure 17. Closed-loop output of the investigated servo system, where a small lag is visible. All poles are placed at −150.

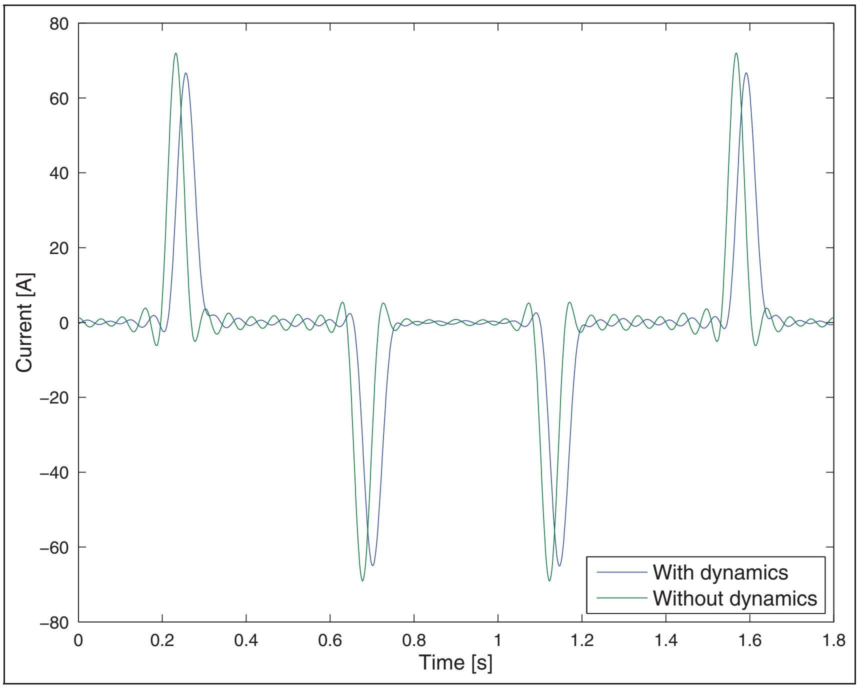

Figure 18. Effort of the actuator for dimensioning with respect to both static property and dynamic behaviour.

closed-loop system might have slower dynamics than the reference profile and hence require less actuation power.

As mentioned in the introduction, it is of high importance that the overall optimisation is fast so that the user does not lose focus on his or her task. Even with a fairly computationally inefficient code, the optimisation process takes only about 80 s for a system

Table 1. Calculated motor current for the case of static property and dynamic behaviour, respectively.

	RMS current (A)	Peak current (A)
Static	17.05656	66.1366
Dynamic	17.0491	61.2874

RMS: root mean square.

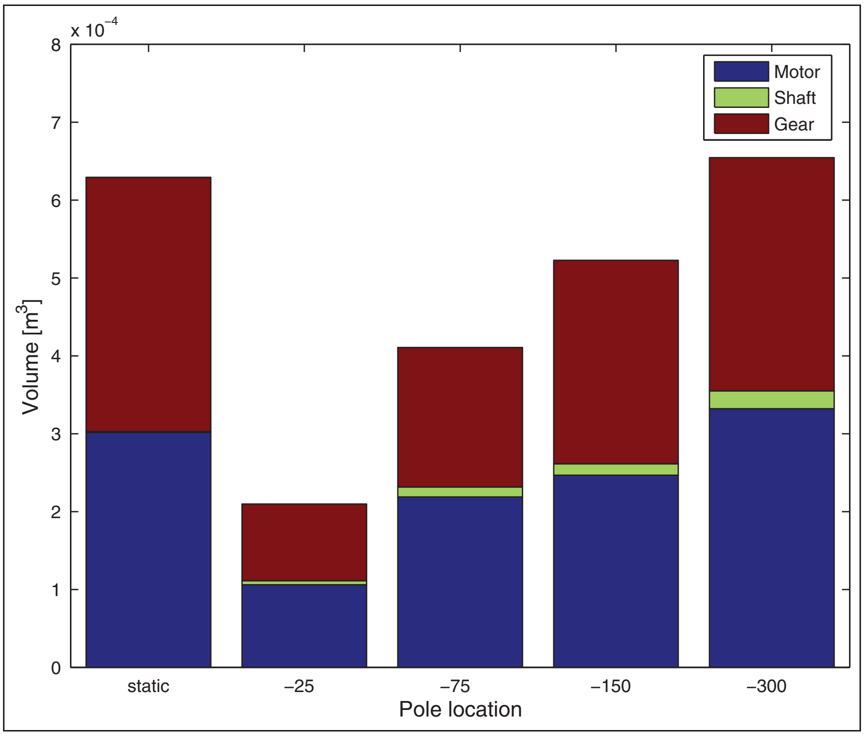

Figure 19. Resulting optimal system volume for different controller specifications. For every pole location bar, all poles were placed at the same location.

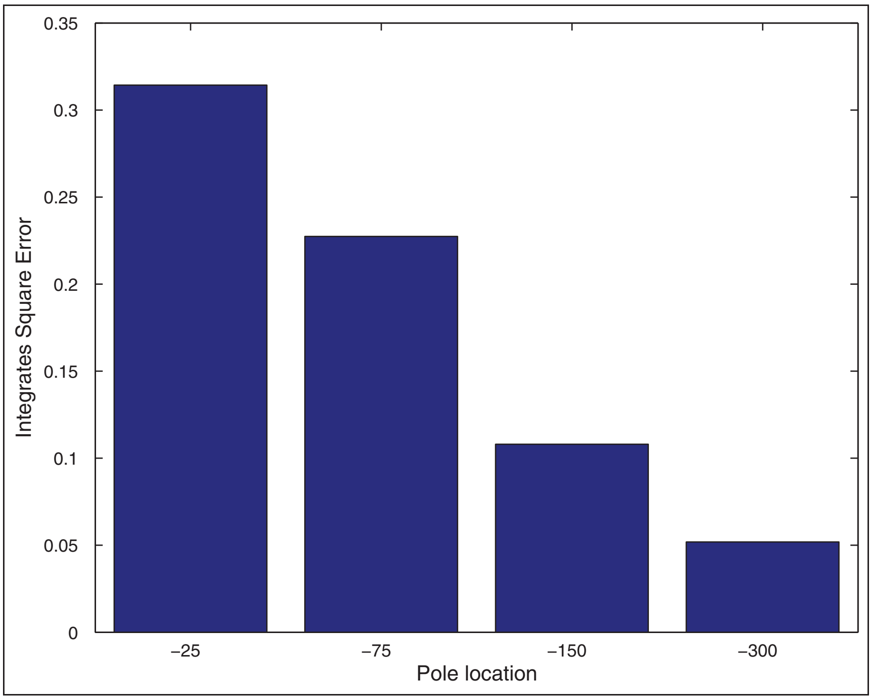

Figure 20. Integrated square error for different controller specifications.

with three components and four optimised design variables. The time required to optimise mechatronic systems of this size can hence be regarded as good enough. However, the authors strongly believe that huge improvements can be achieved through additional improvement of the efficiency of the implementation. This could prove useful if the complexity of the optimised system is increased.

The design case used may seem oversimplified, especially since some component models such as sensor and power circuitry are excluded. It should also be noted that a model of gear stiffness should probably be implemented since it would affect the dynamic behaviour and hence the design case results. However, the method and framework are generic enough to handle any type and number of components as long as suitable static dimensioning and dynamic behaviour models can be derived. Hence, the presented results provide a basis for a tool-supported mechatronic design methodology by which rather complex mechatronic systems can be formally modelled, optimised and evaluated at a very early stage of product design.

It can be concluded that the methodology has good potential, especially with the new addition of dynamic analysis, and as this and previous articles have shown, the hypothesis leading up to this work should now be regarded as more than just a hypothesis.

Ideas for future work

There are a number of interesting paths to investigate for further development of the method and tool. An obvious one would be to do a proper design case from concept to product – or at least from concept to prototype – to further validate the relevance and accuracy of the design methodology. It could also be of interest to investigate different methods to automatically determine suitable pole locations instead of having the user specify them. Another possible path would be to treat the pole locations as design variables for the optimiser and use the ISE of the closed-loop system as an additional optimisation objective. This ISE could be calculated very quickly with the results from the Fourier analysis of the reference profile. Further work should also be put into investigating the effects of introducing friction in more of the dynamic models and in power loss models, since this would likely lead to interesting results. To make the current set of component models more complete, it would be natural to further develop and include some sensor and driver models. The control design aspect currently included is limited to path following so an obvious next step is to include also aspects of disturbance and noise rejection utilising the same 2-degree-of-freedom controller, as shown in Figure 7. Applying the polynomial pole placement approach, this introduces further constraints on $R(s)$ and $S(s)$, depending on the type of disturbance being active.

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

Funding

This research was funded by the Swedish Foundation for Strategic Research through the ProViking program (project no. V08.02).

References

1. VDI 2206. *Design methodology for mechatronic systems*. Düsseldorf: VDI-Verlag, 2004.
2. Li Q, Zhang WJ and Chen L. Design for control: a concurrent engineering approach for mechatronic systems design. *IEEE: ASME T Mech* 2001; 6(2): 161–169.
3. Ulrich KT and Eppinger SD. *Product design and development*. 5th ed.New York: McGraw-Hill/Irwin, 2011.
4. Pahl G, Beitz W, Grote KH, et al. *Engineering design: a systematic approach* (ed Wallace K and Blessing L). London: Springer-Verlag, 2007.
5. VDI 2221. *Systematic approach to the development and design of technical systems and products*. Düsseldorf: VDI-Verlag, 1993.
6. Malmquist D, Frede D and Wikander J. Optimization of mechatronic systems for supporting early design decisions. In: *Proceedings of the 13th mechatronics forum international conference* (ed Scheidl R and Jakoby B), Linz, 17–19 September 2012, pp.490–497. Linz: Trauner Verlag.
7. Malmquist D, Wikander J and Frede D. Optimal design of harmonic drive servo. In: *Proceedings of the 2013 IEEE/ASME international conference on advanced intelligent mechatronics (AIM)*, Wollongong, NSW, Australia, 9–12 July 2013, pp.1579–1584. New York: IEEE.
8. Pil AC and Asada HH. Integrated structure/control design of mechatronic systems using a recursive experimental optimization method. *IEEE: ASME T Mech* 1996; 1(3): 191–203.
9. Savant SV and Asada HH. Integrated structure/control design based on model validity and robustness margin. In: *Proceedings of the 1999 American control conference*, vol. 4, 2–4 June 1999, San Diego, CA, 1999, pp.2871–2875. New York: IEEE.
10. Kajiwara I and Nagamatsu A. Integrated design of structure and control system considering performance and stability. In: *Proceedings of the 1999 IEEE international conference on control applications*, vol. 1, 22–27 August 1999, Kohala Coast, HI, 1999, pp.86–91. New York: IEEE.
11. Da Silva MM, Desmet W and Van Brussel H. Design of mechatronic systems with configuration-dependent dynamics: simulation and optimization. *IEEE: ASME T Mech* 2008; 13(6): 638–646.
12. Da Silva MM, Brüls O, Desmet W, et al. Integrated structure and control design for mechatronic systems with configuration-dependent dynamics. *Mechatronics* 2009; 19(6): 1016–1025.
13. Roos F. *Towards a methodology for integrated design of mechatronic servo systems*. PhD Thesis, KTH Royal Institute of Technology, Stockholm, 2007.
14. Frede D, Malmquist D and Wikander J. Holistic design optimization in mechatronics. In: *Proceedings of the 6th IFAC symposium on mechatronic systems*, Hangzhou, China, 10–12 April 2013, pp.655–662. International Federation of Automatic Control (IFAC), Laxenburg, Austria.
15. Janschek K. *Mechatronic systems design: methods, models, concepts*. Berlin, Heidelberg: Springer-Verlag, 2012.
16. Lenk A, Ballas RG, Werthschützky R, et al. *Electromechanical systems in microtechnology and mechatronics*. Berlin, Heidelberg: Springer-Verlag, 2011.
17. Åström KJ and Wittenmark B. *Computer-controlled systems: theory and design* (ed Hall P). 3rd ed.Upper Saddle River, NJ: Prentice Hall, 1997.
18. Eiben AE and Smith JE. *Introduction to evolutionary computing*. Berlin, Heidelberg: Springer-Verlag, 2003.

19. Roos F, Johansson H and Wikander J. Optimal design of motor and gear ratio in mechatronic systems. In: *Proceedings of the 3rd IFAC symposium on mechatronic systems*, Sydney, NSW, Australia, 6–8 September 2004, pp.1–6. International Federation of Automatic Control (IFAC), Laxenburg, Austria.
20. SS 1863. *Spur gear – geometrical data*. 4th ed.Stockholm: SIS, 1978.
21. SS 1871. *Spur and helical gears – calculation of load capacity*. 3rd ed.Stockholm: SIS, 1978.
22. Collins JA. *Mechanical design of machine elements and machines: a failure prevention perspective*. 1st ed. New York: John Wiley & Sons, 2002.

Appendix 1

Notation

A	denominator of G_p
A_{cl}	closed-loop polynomial
A_d	desired closed-loop polynomial
A_m	polynomial of order equal to A
A_o	polynomial of order equal to the order of A_{cl} minus the order of A_m
b	total width of gearing
B	numerator of G_p
C_m	motor type–specific constant for same cooling conditions
C_{mj}	motor type–specific constant
d	shaft damping
E	Young's modulus
G	shear modulus
G_c	feedback controller transfer function
G_{ff}	feedforward controller transfer function
G_p	open-loop process transfer function
G_{ur}	closed-loop system transfer function from reference to actuator input
G_{yr}	closed-loop system transfer function from reference to output position
i	electrical current
J_g	gear inertia
J_l	load inertia
J_m	motor inertia
J_s	shaft inertia
k	shaft stiffness
K_t	motor torque constant
$K_{H\alpha}$	factor describing division of load between teeth
$K_{H\beta}$	load distribution factor for Hertzian pressure
l	shaft length
l_m	motor's rotor length
n	gear ratio
$p_{1,\ldots,4}$	desired pole locations
r	controller reference
r_g	gearbox outer gear radius
r_i	controller parameters ($i = 0, 1, \ldots$)
r_m	motor stator radius
r_s	shaft radius
R	denominator of G_c
s	Laplace (complex) frequency variable
S	numerator of G_c
S_i	controller parameters ($i = 0, 1, \ldots$)
t_0	static gain that gives unit DC gain in the closed-loop transfer function
T	feedforward polynomial
$T_{g,in}$	gearbox input torque
$T_{g,out}$	gearbox output torque
$\tilde{T}_{g,out}$	gearbox output RMS torque
$T_{l,in}$	load input torque
$T_{l,out}$	load output torque
$\tilde{T}_m$	motor RMS torque
$T_{m,out}$	motor output torque
$T_{m,rated}$	motor rated torque
$\hat{T}_s$	shaft peak torque
$T_{s,in}$	shaft input torque
$T_{s,out}$	shaft output torque
V_g	gearbox volume
y	system output
Z_H	form factor for Hertzian pressure
Z_M	material factor for Hertzian pressure
Z_α	contact ratio for Hertzian pressure
η	gear efficient
ν	Poisson's ratio
$\sigma_{H,max}$	maximum allowed flank pressure
τ	cycle time of the load profile
$\tau_{s,max}$	maximum permissible shear stress
$\varphi_{g,in}$	gear input angular position
$\varphi_{g,out}$	gear output angular position
$\varphi_{l,in}$	load input angular position
$\varphi_{l,out}$	load output angular position
$\varphi_{m,out}$	motor angular position
$\varphi_{s,in}$	shaft input angular position
$\varphi_{s,out}$	shaft output angular position