Quasi-Periodic Feedforward Control Based on Inverse Model Tabled FFT

Tilman Bünte[1] Jakub Tobolář[2]

[1,2]Institute of Vehicle Concepts, German Aerospace Center (DLR), {Tilman.Buente, Jakub Tobolar}@dlr.de

Abstract

Mitigating periodic oscillations (e.g. in rotating systems) is a common control engineering problem. Fast Fourier Transform (FFT)-based methods are well-suited for respective analysis. While FFT algorithms inherently assume signal periodicity, rotating systems often exhibit true periodic behavior (e.g., shaft rotation frequencies). Using angle-sampled data rather than time-sampled data allows direct analysis of oscillations relative to rotational cycles, which is particularly useful for tracking unbalance or periodic external excitation in rotating assemblies.

Modelica provides several built-in resources to address these challenges. First of all, inverse models have the potential to derive an ideal control signal in time domain. For periodic disturbances, this ideal control is likely to be approximated well by a periodic, i.e. Fourier-transformable signal. Modelica is an appropriate model environment to store and retrieve tabled FFT data depending on operating conditions such as rotational speed. In a real-time application, synthesizing control signals from pre-computed Fourier tables offers a practical alternative to executing potentially complex inverse models online, reducing computational effort and system complexity. The paper demonstrates this approach using the example of mitigating oscillations induced by an internal combustion engine in a hybrid automotive power train.

Keywords: Model Inversion, Fast Fourier Transform, Fourier Synthesis, Vehicle Power Train, Torsional Vibration Suppression, Disturbance Cancelling

1 Introduction

This paper addresses mitigation or cancellation of periodic or quasi-periodic oscillations in technical systems caused by external or self-excitation mechanisms. To ensure clarity and simplicity, the presentation here focusses on mechanical 1D-rotational systems. However, the results can analogously be applied to other technical fields, for instance electric circuits or acoustic systems. The presence of a suitable actuator such as an electric motor is assumed which can be used as a torque source for control action. Considerations are here restricted to feedforward control without denying the potential of feedback control. More precisely, the approach presented here could well be combined with additional feedback control or other advanced methods like iterative control, invariant manifold tracking, and hybrid testing.

The novelty and main contribution of this paper is the Modelica based combination of the techniques

a) of model inversion for the synthesis of a theoretically perfect feedforward control,

b) the Short-Time Fourier Transform (STFT) of the quasi-periodic control signal and

c) model reduction of the control by approximating it with a correspondingly Fourier synthesized signal.

Moreover, a Modelica package is introduced and a practical application is demonstrated. To sum up the corresponding state of the art:

a) Model inversion in Modelica is a well established technique (Thümmel et al. 2005).

b) STFT is primarily used for analysing non-stationary signals whose frequency spectrum changes over time. In (Bünte 2011) it is used to determine describing functions of Modelica models. In (Kuhn 2017) it is used for stationary periodic steady state identification of AC electrical systems.

c) FFT for model reduction is applied for instance in the fields of material science (Gierden 2024) and Finite Element Method (FEM) based product design (Hülsebrock 2023).

In addressing the issue of quasi-periodic disturbances impacting a rotational system, we consider that the primary period of any given oscillation maintains a constant ratio relative to a single revolution of a reference shaft. For instance, an internal combustion engine (ICE) generates a periodic torque and hence torsional oscillations in the drivetrain due to the sequential firing of its cylinders. The harmonic structure of the crankshaft torque oscillations is typically influenced by factors like ICE speed, load torque, and throttle position. Given the known period of the oscillation, for a specific operational case (ICE speed, throttle position etc.) a STFT of the periodic crankshaft torque can be performed over the time window of one period. The resulting Fourier coefficients for a limited number of harmonics approach the dynamic torque characteristic of the ICE for this operational case. Thus, the complex thermodynamic process plus piston and rod related

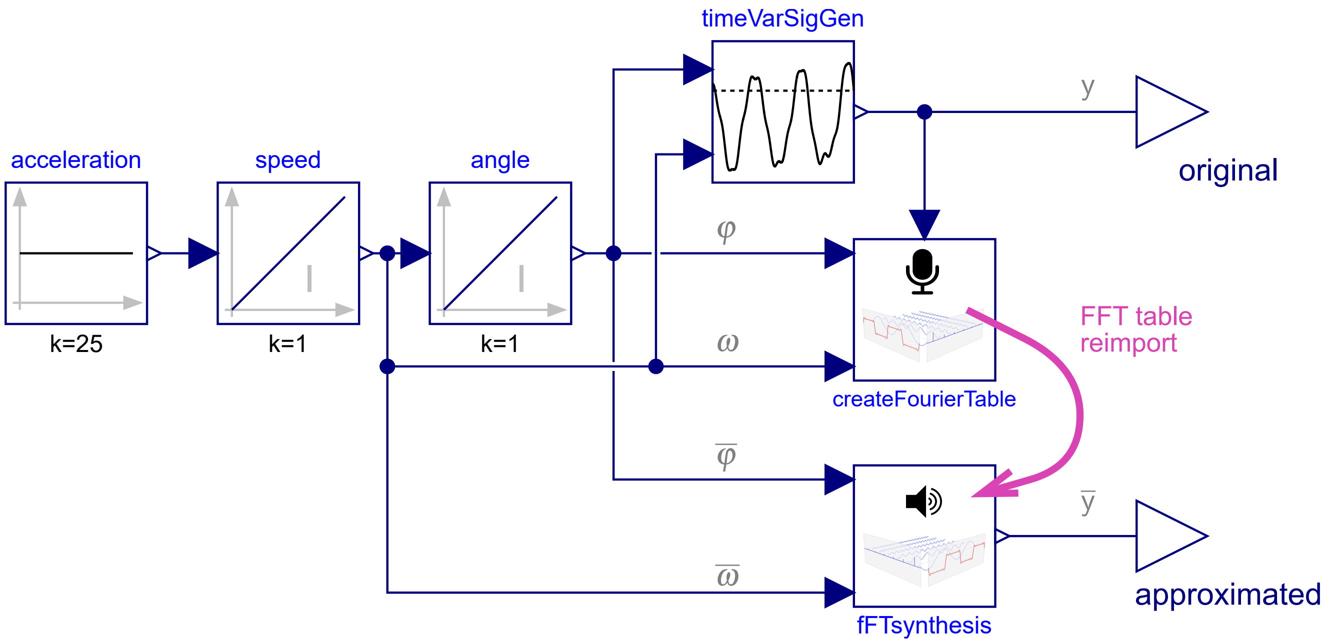

Figure 1. Demo Modelica model combining instances of the blocks `CreateFourierTable` and `FFTsynthesis`.

inertia dynamics in an ICE could be approximated and represented by a set of Fourier coefficients for the torque on the crankshaft (Chauvin et al. 2004), stored in a (multidimensional) table depending on operational parameters while using appropriate gridding. Obviously, a Modelica model of an ICE can be easily built as a Fourier synthesis-based signal generator acting as torque source and using the above-mentioned table for parameterization.

The paper is organized as follows. Section 2 presents a concise Modelica package which comprises classes for a) the repeated determination of STFT coefficients for a signal of a time varying oscillating system and b) a signal synthesizer as an effective approximative representation of the previously decomposed system and c) a demo model combining a) and b). A complex quasi-periodic nonlinear system which may be difficult to be solved numerically can thus potentially be replaced by interpolation from a lookup table. For some cases this can serve as a model reduction technique e.g. in view of real time capability. Section 3 considers the cancellation of quasi-periodic disturbances on a system by means of model inversion (well familiar to Modelicans (Thümmel et al. 2005)). Section 4 integrates the features of Sections 2 and Section 3 and thus, represents the main contribution of this work: A feedforward control scheme is presented which is based on the STFT decomposed ideal control signal from the inverse model. In Section 5, a hybrid automotive power train model with ICE and integrated starter generator (ISG) is used to demonstrate the practical applicability and relevance of the feedforward control scheme. Finally, Section 6 provides a conclusion and an outlook.

2 A Modelica Package for Fourier Decomposition and Playback of Quasi-Periodic signals

A Modelica package `FourierTable` was designed which provides

- a recording block `CreateFourierTable` using repeated STFT of a quasi-periodic signal y occurring in a Modelica model and
- a counterpart block `FFTsynthesis` for playback of a signal $\bar{y}$ using Fourier synthesis.

See Fig. 1 for an example which combines both classes in one Modelica model. A reference y ("original") is generated by a time varying signal generator `TimeVarSigGen`. The simulation of this demo model allows for direct comparison of y and its approximation $\bar{y}$. To do this, it is necessary to simulate twice. In the first run, the Fourier coefficient table is generated by `CreateFourierTable` and saved to a mat-file. During initialisation of the second run, this table is loaded as parameterization of `FFTsynthesis` which facilitates a meaningful comparison.

For the recording using `CreateFourierTable` it is assumed that the total Modelica model has the following features:

- There is a signal ω and its integral φ (e.g. with the physical meaning of angular speed and position, respectively, of a rotating shaft). It is further assumed that ω is varying slowly enough, such that “quasi-periodicity” of a considered signal y is always sufficiently well maintained. The reader is referred to Section 2.4 for deeper considerations.

DOI 10.3384/ecp218619

- The actual fundamental period of the signal y corresponds to an increase of φ by 2π (one revolution of the shaft). Consequently, periodicity of y relates to φ, not time.
- The composition of the signal y in terms of harmonics depends only on one single time varying parameter q. The latter could be but not necessarily is $q = \omega$. For the sake of simplicity, this case is assumed in the following.

During a simulation of the considered system while typically ω is monotonously increasing, events are triggered when ω reaches predefined threshold values. After each event, a STFT analysis of y is performed over the window of a complete period of φ (i.e. one revolution). Hereafter, the Fourier coefficients are stored into a new row of a table with the actual ω value in the first column, cf. Table 1 for illustration.

For playback (normally in another Modelica model), the previously collected tabled data is used as parameterization for an instance of `FFTsynthesis`. Using $\overline{\varphi}$ and $\overline{\omega}$ as inputs to the playback block, Fourier coefficients are interpolated from the table and a quasi-periodic signal $\overline{y}$ is generated which approaches y of the original system for $\overline{\omega} = \omega$ and $\overline{\varphi} \pmod{2\pi} = \varphi \pmod{2\pi}$.

The record and playback blocks are now described in more detail.

2.1 Data Sampling and Repeated STFT of the Quasi-Periodic Signal

The block `CreateFourierTable` has the following parameters:

```
parameter Integer n = 9 "Exponent; Number
 of angle samples per period is np = 2^n";
parameter Modelica.Units.SI.AngularVelocity
  wnodes[:] = linspace(10,100,5) "List of
 frequency nodes where data acquisition for
 STFT evaluation is triggered";
parameter Integer nh = 2 "Number of
 harmonics";
```

The block mainly consists of a single algorithm section. It is activated each time when the monotonously increasing signal ω surpasses one of the scheduled `wnodes` values. Now, sampled data for y is being collected during the newly started period in order to prepare for later STFT. The current φ period is divided into an equidistant grid of 2^n samples according to the model parameter `n` and data collection is triggered at each φ grid point. After the period is complete, the block is put to sleep til the next ω-induced activation in order to save non-essential events and thus to reduce computational effort.

As soon as data collection over the period is completed, an STFT is performed using the function `Math.FastFourierTransform.realFFT()` (Cooley and Tukey 1965) from Modelica Standard Library. A second input to the function is the number of frequencies specified by the model parameter `nfi` $= 1 +$ `nh` equalling the DC part plus the desired number of harmonics to be considered. The resulting amplitudes a_i and phases ϕ_i of the harmonics are stored into a new row of a table whose first column holds the mean value of ω during the collected period, cf. Table 1. Hence, the ω node differs a bit from the actual `wnodes` value. In most cases the exact values of the table interpolation nodes does not matter; the parameterization of `wnodes` can be used for an approximate scheduling. The table is filled during the simulation of the total model while ω is monotonously increasing and sweeps over the range of scheduled `wnodes` values. Section 2.4 will provide comments on selecting `wnodes` and appropriately scheduling the rate of change for ω.

Upon end of simulation, the resulting table is stored as a matrix which represents the set of Fourier coefficients for the quasi-periodic signal y at the given ω nodes. The sampled data collection of quasi-periodic signals and Fourier decomposition was reused from (Bünte 2011).

2.2 Signal Approximation by Fourier Synthesis

A set of Fourier coefficients (compliant with the format of the table generated from `CreateFourierTable` as described before) is the only parameter of the block `FFTsynthesis`:

```
parameter Real table[:,:] "FFT table as
 function of angular velocity (w rev/s (col.
  1), amplitudes (col. 2:nfi+1), phases (
 col. nfi+2:end))";
```

For the given input value of $\overline{\omega}$ (cf. Fig. 1), the corresponding set of Fourier coefficients is interpolated from the table. The synthesis of the signal $\overline{y}$ is straightforwardly computed by superposition of the `nfi` frequencies (DC part plus `nh` harmonics) with the (real or virtual shaft angle) input $\overline{\varphi}$ as independent variable:

$$\overline{y}(\overline{\varphi}, \overline{\omega}) = \sum_{i=0}^{nh} a_i(\overline{\omega}) \cos\left(i\,\overline{\varphi} + \phi_i(\overline{\omega})\right) \tag{1}$$

For a constant $\overline{\omega}$ the resulting signal $\overline{y}$ is strictly periodic. If $\overline{\omega}$ is slowly varying compared to the periodicity 2π of the cosine function argument, then $\overline{y}$ renders *quasi-periodic*.

2.3 Demo Model Simulations

Fig. 2 presents the results of two successive simulations of the demo model introduced with Fig. 1. The *original* output from the signal generator `TimeVarSigGen` consists of a DC part plus a stationary first harmonic plus a second harmonic whose amplitude changes with ω. Finally, there is a stationary fifth harmonic with a small amplitude:

```
y = -2.212
    +3.346*cos(1*phi + 0.526)
    +1.233*cos(2*phi + 0.745)*(0.3+1*cos
     ((w-10)/5))
    +0.300*cos(5*phi - 0.745);
```

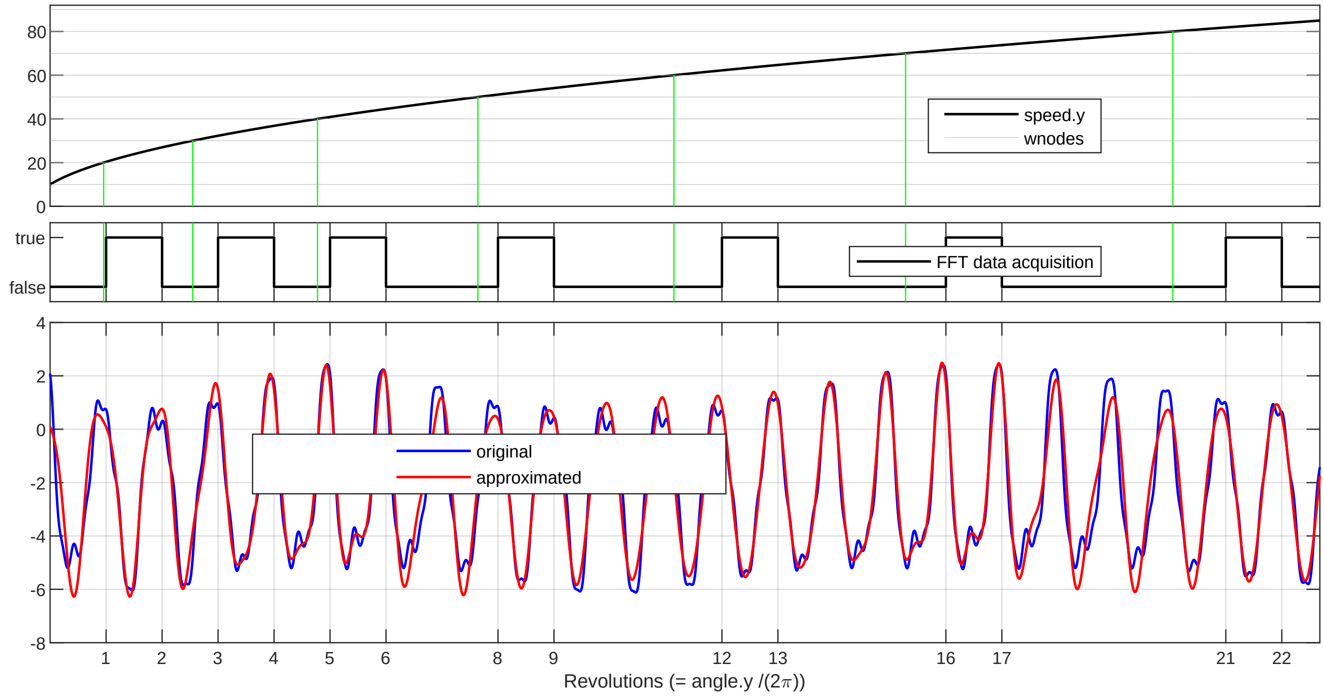

Figure 2. Simulation results from the demo Modelica model shown in Fig. 1. Top: Rotational speed and thresholds; Middle: Indication of data acquisition phases; Bottom: Comparison of original signal y with its approximation $\bar{y}$.

A first run of the model is made here with `CreateFourierTable.nh` $= 2$, i.e. only two harmonics are to be considered for the STFT. The table generated by `CreateFourierTable` is automatically saved to a file and then used as parameterization for the block `FFTsynthesis` in a second simulation. (Additional simulations with the model would deliver the same results as from the second run.) The upper plot in Fig. 2 shows the rising of $\omega =$ `speed.y` and the constant `wnodes` thresholds. Green vertical lines indicate the instants where the thresholds are surpassed. The FFT data acquisition phases, each covering the complete subsequent revolution, can be seen from the middle plot. Rather than time, in all plots the number of revolutions equalling $\varphi/(2\pi)$ is used as abscissa. The lower plot compares the *original* signal `TimeVarSigGen.y` with the *approximated* signal `FFTsynthesis.y`. Table 1 exhibits the example's numerical values for the Fourier coefficients interpolation table, compliant with the notation used in eq. (1). Where $\overline{\omega}$ is close to interpolation nodes, the lower plot shows good agreement of the approximated signal $\bar{y}$ (red) with the original reference y (blue). Understandably, here the interpolation of the Fourier coefficients for signal reconstruction matches best the recorded characteristics.

2.4 Parameters Influencing Approximation Quality

Obviously, the accuracy of the approximated signal depends on some parameters during data acquisition. Some skill and experience is required for their choice. In practice, the parameters of `CreateFourierTable` (see Section 2.1) and the total model can be adapted such that sufficient precision of the approximated signal is attained:

1. If denser gridding of `wnodes` is used, then the agreement of $\bar{y}$ with y is correspondingly better in-between

Table 1. Interpolation table for Fourier coefficients, compliant with the notation used in eq. (1), resulting from the demo model simulations.

$\overline{\omega}$	a_0	a_1	a_2	$\phi_0[°]$	$\phi_1[°]$	$\phi_2[°]$
23.827	−2.2350	3.3433	0.71776	0.0000	31.639	−136.93
34.612	−2.1557	3.4258	0.65922	0.0000	27.589	42.980
42.742	−2.2273	3.3041	1.5210	0.0000	30.678	42.697
52.629	−2.2416	3.2993	0.40789	0.0000	31.544	−137.03
63.456	−2.1803	3.3917	0.026966	0.0000	28.770	45.790
72.687	−2.2122	3.3350	1.5903	0.0000	30.089	42.699
82.791	−2.2347	3.3057	0.14749	0.0000	31.174	−136.82

DOI 10.3384/ecp218619

the interpolation points.

2. The same applies to the rate of change of ω during data acquisition. If ω or any other operating conditions q are not changing slowly enough, periodicity is not sufficiently satisfied and the result may be spoiled. Generic statements about reasonable limits on the rate of change are difficult to derive and exceed the objective of this publication. A heuristic approach will serve well in most applications and the rise of ω should be controlled to be sufficiently moderate in the simulation such that the results are satisfactory.

3. The number `nh` of harmonics considered in the recording affects the achievable accuracy and should be chosen accordingly.

4. A large value of the parameter `n` contributes to good results. The Nyquist–Shannon sampling theorem (Franklin et al. 1990) should be regarded in the context of the considered harmonics. On the other hand, more data sampling events are triggered in the block `CreateFourierTable` during the FFT data acquisition phases and the analysis simulation takes more effort.

Note that in the example shown in Fig. 2 some of the above enumerated recommendations are violated for the sake of illustrating their effect: Due to the rapid rise of ω the periodicity assumption was not held. As a side effect, the resulting set of interpolation nodes for ω (cf. Table1) is coarse. The STFT only considered two harmonics, therefore the fifth harmonic oscillation of the original signal is not captured.

3 Inverse Model Based Disturbance Cancellation

Here, we use the term *disturbance* to mean an undesired effect on a specific system signal induced by (quasi-)periodic external or self-excited quantities.
Inverse linear and nonlinear models are of significant meaning as building blocks for the design of control systems, both for feedforward and feedback control. One of the outstanding features of Modelica is that nonlinear inverse models may under circumstances be automatically generated from the plant model (Thümmel et al. 2005).
Particularly, when addressing the problem class considered in this paper, the ideal control signal for cancelling disturbance can be derived using an inverted model, provided that the conditions for model inversion are fulfilled such as uniqueness and stable zero dynamics. Details on this can be found in (Thümmel et al. 2005; Thümmel 2007). To accomplish model inversion, the equation defining the control signal (i.e. the actuator input) is removed. Hence the total model equations will be solved for this quantity to be an output. The number of equations needs to remain balanced with the number of unknowns such that the model equations can be solved and the model can be translated and simulated. Therefore, replacing the removed control signal, an equation is added to the Modelica model which represents a desired behaviour of a specific model's quantity. In our case, this is an adequate disturbance cancelling condition, e.g. with the meaning that a desired target signal is set to zero or follows perfectly a certain reference.

In the case that the inverse model is lean and real-time capable, it can be used to generate the control signal for a given (periodic) disturbance and that control signal can be applied feedforward to a real system. If the basic model matches well enough the real system, then the control signal has the potential to significantly mitigate the disturbance. However, in general, solving the inverse model in real time may be unfeasible or computationally expensive. Moreover, the approach presupposes that the inverse model stays in tune with the real system in terms of system states and/or operating conditions. These challenges suggest to use an abstraction of the inverse model which combines the approaches from Section 2 and this one, to be introduced in the next section.

4 Feedforward Disturbance Cancellation Using Fourier Synthesized Approximation of the Ideal Control Signal

As this section is the core, the key idea of the paper is presented in four subsequent steps while using the Modelica implementation of the recording and playback blocks presented in Section 2:

1. As outlined in the previous section, an inverse model can be employed to generate an ideal, theoretic control signal for a given objective such as cancelling a target signal. In case of constant operating conditions including a perfectly periodic disturbance and the system model being in steady state oscillation this approach yields a periodic solution for the control variable.

2. The *pattern* of this periodic signal in terms of its harmonics can be expressed as Fourier coefficients and be considered as an eligible blueprint for periodic feedforward control action (to be applied with appropriate phase) for the given system model at steady operating conditions.

3. The blueprint can be applied to a real system by means of a Fourier synthesizer if
   - the periodic trigger for the disturbance can be measured or estimated and
   - the system matches the operating conditions of the underlying model which was used to generate the pattern.

4. If the operating conditions of the real system do vary and it is not in perfect steady state, then modified patterns are applied, each fitting the prevailing operating conditions.

    - For a manifold of operating conditions, the patterns are Fourier synthesized based on interpolated Fourier coefficient datasets which were generated each via STFT using a sufficiently dense node gridding to achieve good approximation.
    - For practical application of the approach, the recording of the patterns with the inverse system model should be done while the operating conditions are varying sufficiently slowly such that the inverse system model is always in quasi steady state.
    - The same applies to the playback of the patterns on the real system. The results will be best if the operating conditions vary only slowly such that the real system model is in quasi steady state oscillation.

5 Automotive Application Example: Hybrid Power Train

The application example used here is based on a simplified automotive mild hybrid power train consisting of a four-stroke four-cylinder internal combustion engine (ICE) plus integrated starter generator (ISG) mounted on the extension of the crankshaft.

5.1 Modelica Power Train Uncontrolled Base Model

The total Modelica model of the power train, depicted in Fig. 4, is completed by an abstracted drive train including rotational spring and damper plus an inertia representing the total load including the translated mass of the vehicle body. To facilitate a perfectly common scenario for various variants to be compared later, the model features a signal source (`speedScheduler`) which imposes a predesigned rotational speed profile to the inertia.
The ICE produces a quasi-periodic dynamic torque that varies with rotational speed. This torque consists of two main components:

- The forces generated by the pressure in the four cylinders, which follow the firing order and occur every second crankshaft revolution, influenced by the geometric arrangement like in boxer or V engines (Laschet 1988).
- The asymmetric inertial forces of the pistons and connecting rods.

Fig. 3 illustrates the resulting periodic torque pattern at full throttle for the ICE model used with a set of curves representing different crankshaft speeds.

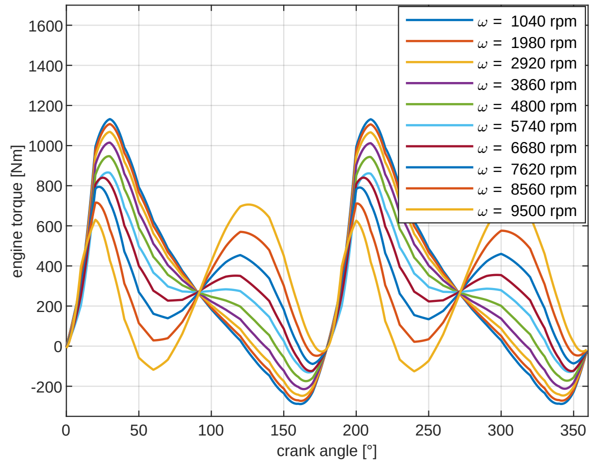

Figure 3. Full throttle periodic output torque of the four-stroke four-cylinder ICE depending on crank shaft speed ω.

The objective of the feedforward control derived in the next two subsections is to compensate for the quasi-periodic ICE torque variations around the DC torque component of the ICE such that torsional oscillations in the drive train are minimized.

5.2 Fourier Coefficient Recording of Ideal Control Using Inverse Model

Following the concept introduced in Section 2.1, a Fourier coefficient table is generated using the inverted model shown in Fig. 5. Therefore, starting from the model of the uncontrolled base model (Fig. 4), a `CreateFourierTable` block is added to the system. The block `Math.InverseBlockConstraints` from the Modelica Standard Library is used to invert the model such that the combined output torque of ICE plus ISG equals a constant torque (set to 280 Nm here to reflect the approximate mean torque of the ICE). In return, the equation defining the input torque of the ISG is dropped. By effect of the model inversion this quantity is calculated during the simulation and, repetitively, the STFT is performed after discrete rotational speed values are surpassed.

5.3 Power Train Feedforward Control for Approximative Disturbance Cancellation

Feedforward control of the hybrid power train using the Fourier synthesis introduced in Section 2.2 is applied here. The model shown in Fig. 6 uses the ISG torque as control variable. An instantiation of the block `FFTsynthesis` contains the FFT approximated and tabled information of the perfect disturbance cancellation control previously acquired using Fig. 5 as described above.

5.4 Comparative Simulation Results

For the plots shown in Fig. 7, results are compiled from simulations with the three models described in Sections 5.1-5.3 and Figures 4-6, respectively.

The scenario for all three simulations is the same: The drive train is accelerated constantly according to a given

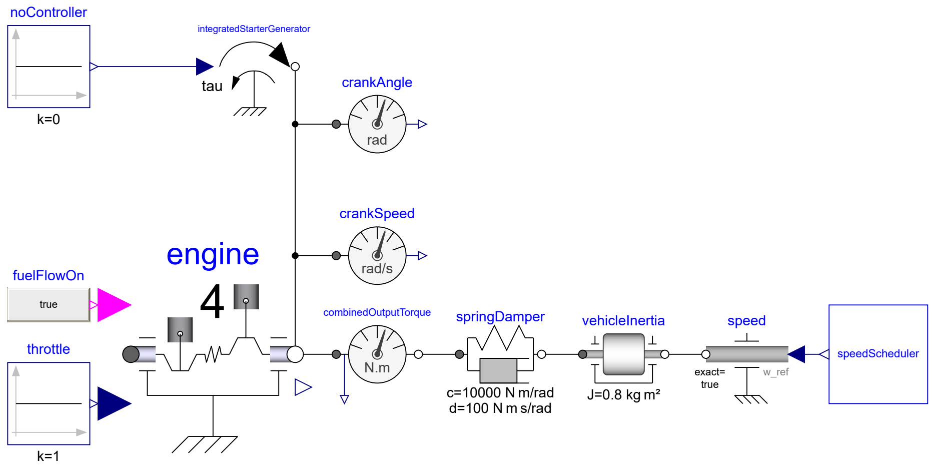

Figure 4. Uncontrolled power train model with imposed speed.

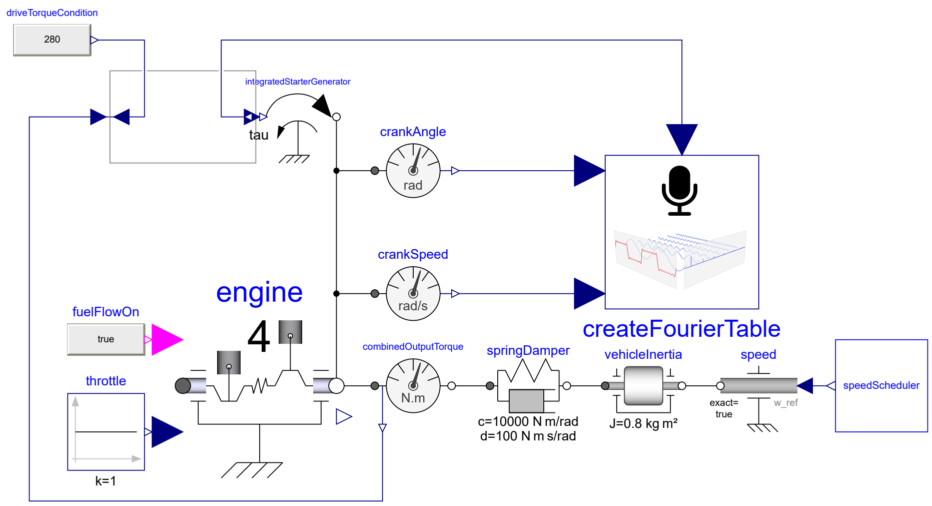

Figure 5. Inverse power train model for repetitive STFT recording of ideal control action.

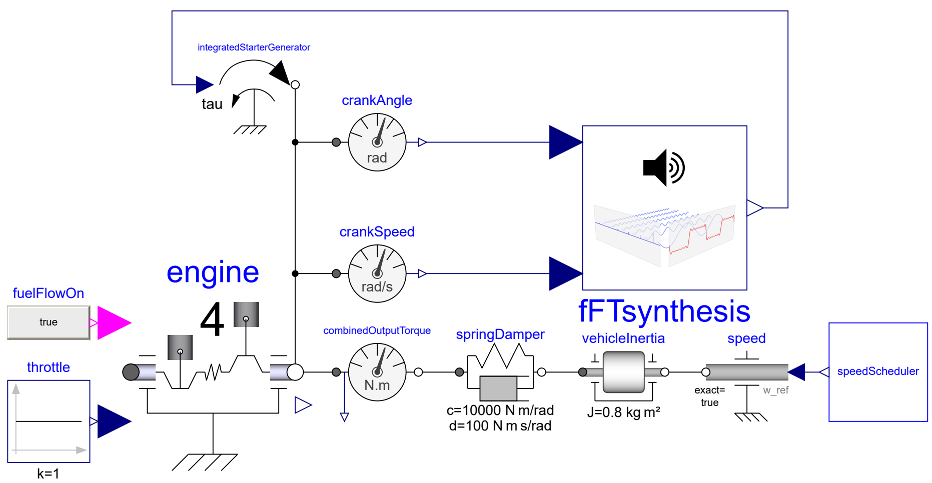

Figure 6. Feedforward controlled power train.

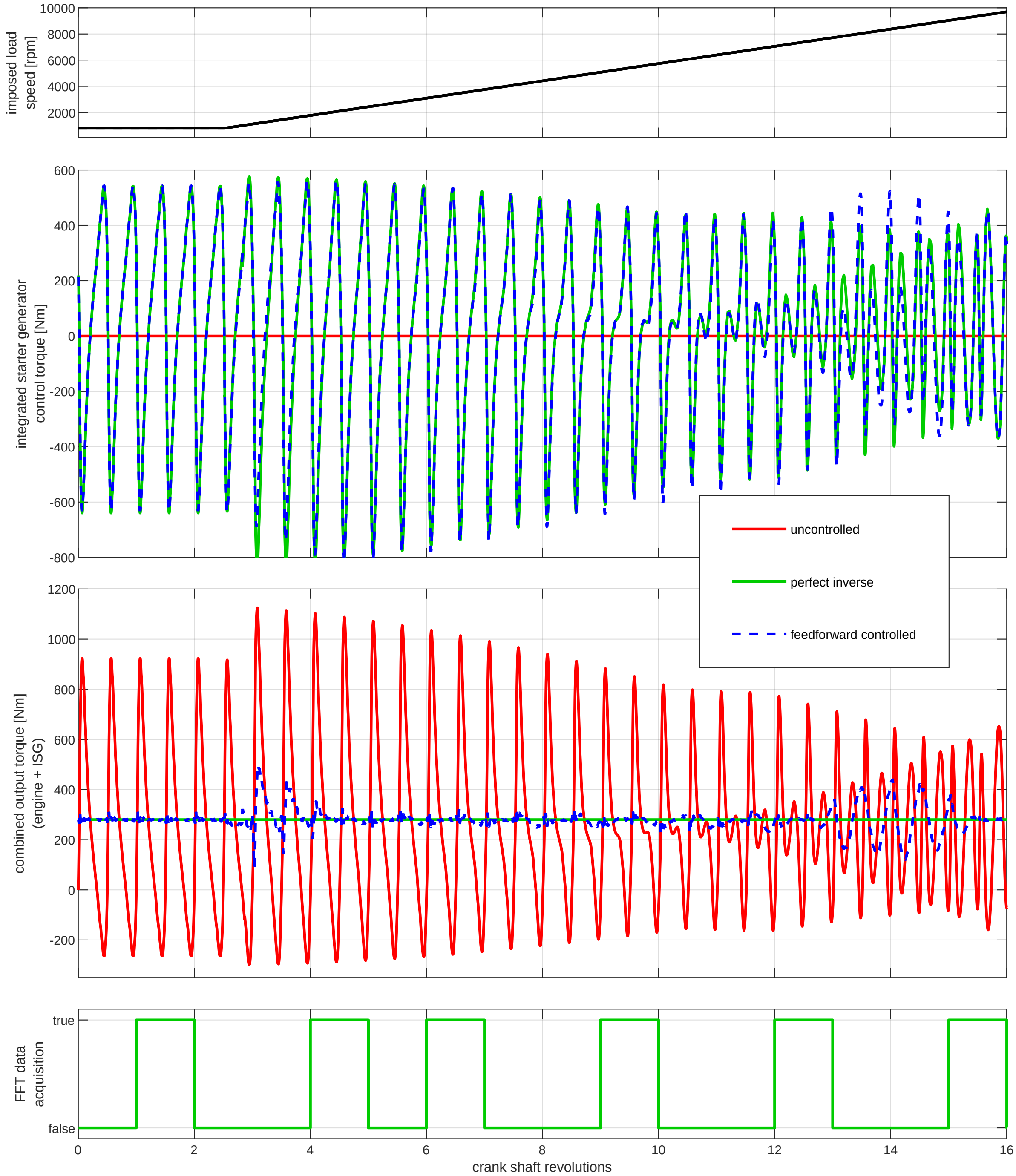

Figure 7. Simulation results comparing various variants of the automotive power train under quasi-periodic disturbance. 1st row plot: rotational speed; 2nd row: ISG control torque; 3rd row: total torque from ICE+ISG; 4th row: data acquisition phases.

DOI 10.3384/ecp218619

speed ramp. By this means, the acceleration is even higher than in a flying start with the purpose that within only few revolutions a large range of speeds is covered. Thus, the speed dependency of the quasi-periodic disturbance is well cognizable in the plots of single simulations. The uppermost plot in Fig. 7 shows the imposed speed of the drive train being the same for all three models. The control input (i.e. the ISG torque) is plotted in the second row plot. The control torque of the uncontrolled base model (corr. to Fig. 4, red line) is zero. The resulting combined output torque of ICE plus ISG exhibits high amplitude quasi-periodic spikes (see third row plot, red line). With model inversion applied (green line each, corr. to Fig. 5) the combined torque is set to zero and the resulting perfect (theoretic) ISG control torque renders quasi-periodic. During the same simulation, data for STFT is acquired in the complete revolutions where the corresponding flag is true, see lowermost plot. After completion of each data acquisition phase, computed Fourier coefficients are stored in the FFT table in a node associated with the mean speed of the past revolution. As usual with interpolation from tabled data, the result is most precise when the query speed is close to one of the speed nodes. This effect can be observed from the simulation results with the feedforward controlled model (corr. to Fig. 6, blue dashed lines). At speeds where FFT data was acquired with the inverse model, the feedforward control performs significantly better in terms of the combined output torque.

6 Conclusion and Outlook

In this paper the combination of the following three Modelica based techniques has been introduced:

- Model inversion for the synthesis of a theoretically perfect feedforward control aiming at compensation of time varying periodic oscillations.
- Repetitive Short-Time Fourier Transform (STFT) of the quasi-periodic control signal.
- Model reduction of the ideal control by approximating it with a correspondingly Fourier synthesized signal depending on varying operation conditions.

A Modelica package has been explicated and the practical application to mitigation of self-excited quasi-periodic disturbances in an automotive hybrid power train was demonstrated.

The effectiveness of the presented feedforward control scheme depends on the periodicity of the disturbances. The operating conditions (for example the rotational speed) should be varying slowly enough such that quasi-periodic behaviour prevails at all times, i.e. the relevant signal should be well approximated in sections by periodic signals. Moreover, a measurement or estimation of the disturbance trigger (e.g. the rotational angle of a shaft) should be available. And, the disturbance to be mitigated should be truly (reproducibly) depending on the trigger. Under these assumptions a feedforward control following the concept described here should be worthwhile to be considered for cancellation of oscillations.

Note that suppression of disturbances which are stochastic (noise) or of other external or non-(quasi-)periodic nature are not in the scope of the approach. Therefore, these were not considered here.

Possible enhancements of the approach include the following:

- The use of more elaborated window functions to mitigate spectral leakage artefacts.
- Currently, the window for data acquisition and Short-Time Fourier Transform covers one shaft revolution, with the shaft angle starting at the next multiple of 2π. An appropriate window shift could handle this issue to get the ω interpolation nodes of the resulting table closer to the scheduled `wnodes` values.
- The Fourier coefficient table employed so far covers a single varying parameter q which was synonymic with ω in this paper. However, a second influencing variable (such as the throttle position in our application example) could be considered and the table could provide bivariate interpolation (using `CombiTable2Ds` instead of `CombiTable1Ds` within `CreateFourierTable`). Of course, this requires an elaborate parameter gridding and approaching schedule (going beyond the previous concept explained with `wnodes`) for the analysis/recording simulation while using the `CreateFourierTable` features.

References

Bünte, Tilman (2011). “Recording of Model Frequency Responses and Describing Functions in Modelica”. In: *8th International Modelica Conference 2011*. URL: https://elib.dlr.de/68920/.

Chauvin, J., G. Corde, P. Moulin, M. Castagne, N. Petit, and P. Rouchon (2004). “Real-time combustion torque estimation on a diesel engine test bench using an adaptive Fourier basis decomposition”. In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. Vol. 2, 1695–1702 Vol.2. DOI: 10.1109/CDC.2004.1430288.

Cooley, James W. and John W. Tukey (1965). “An Algorithm for the Machine Calculation of Complex Fourier Series”. In: *Mathematics of Computation* 19.90, pp. 297–301. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/2003354 (visited on 2025-07-30).

Franklin, Gene F., David J. Powell, and Michael L. Workman (1990). *Digital Control of Dynamic Systems*. Second, pp. xxi + 837. ISBN: 0-201-11938-2.

Gierden, Christian (2024). “Model order reduction for FE-FFT-based multiscale material modeling”. Dissertation. Aachen: Rheinisch-Westfälische Technische Hochschule Aachen. DOI: 10.18154/RWTH-2024-05191.

Hülsebrock, Moritz (2023). "Methode zur parametrischen globalen Modellordnungsreduktion für eine effiziente Validierung dynamischer Systeme". de. PhD thesis. Darmstadt: Technische Universität Darmstadt, xvi, 114 Seiten. DOI: https://doi.org/10.26083/tuprints-00022844.

Kuhn, Martin R. (2017). "Periodic Steady State Identification of electrical circuits". In: *12th International Modelica Conference*. Linköping Electronic Conference Proceedings. Linköping University Electronic Press, pp. 493–505. URL: https://elib.dlr.de/115022/.

Laschet, Andreas (1988). *Simulation von Antriebssystemen: Modellbildung der Schwingungssysteme und Beispiele aus der Antriebstechnik*. Zugl.: Aachen, Techn. Hochschule, Dissertation u. d. T. : Laschet, Andreas: Entwicklung eines Verfahrens zur rechnergestützten Simulation von Torsionsschwingungen in Antriebssystemen. Berlin (de): Springer.

Thümmel, Michael (2007). "Modellbasierte Regelung mit nichtlinearen inversen Systemen und Beobachtern von Robotern mit elastischen Gelenken." Dissertation. Technische Universität München. URL: https://elib.dlr.de/51315/.

Thümmel, Michael, Gertjan Looye, Matthias Kurze, Martin Otter, and Johann Bals (2005). "Nonlinear Inverse Models for Control". In: *Proceedings of the 4th International Modelica Conference*. Ed. by Gerhard Schmitz. LIDO-Berichtsjahr=2005, pp. 267–279. URL: https://elib.dlr.de/12298/.

DOI 10.3384/ecp218619