International Journal of Control, 2015
Vol. 88, No. 7, 1389–1399, http://dx.doi.org/10.1080/00207179.2015.1012558

Taylor & Francis
Taylor & Francis Group

Optimal current waveforms for brushless permanent magnet motors

Nicholas Moehle[a,*] and Stephen Boyd[b]

[a]*Mechanical Engineering Department, Stanford University, Stanford, CA, USA;* [b]*Electrical Engineering Department, Stanford University, Stanford, CA, USA*

(*Received 21 July 2014; accepted 12 January 2015*)

In this paper, we give energy-optimal current waveforms for a permanent magnet synchronous motor that result in a desired average torque. Our formulation generalises previous work by including a general back-electromotive force (EMF) wave shape, voltage and current limits, an arbitrary phase winding connection, a simple eddy current loss model, and a trade-off between power loss and torque ripple. Determining the optimal current waveforms requires solving a small convex optimisation problem. We show how to use the alternating direction method of multipliers to find the optimal current in milliseconds or hundreds of microseconds, depending on the processor used, which allows the possibility of generating optimal waveforms in real time. This allows us to adapt in real time to changes in the operating requirements or in the model, such as a change in resistance with winding temperature, or even gross changes like the failure of one winding. Suboptimal waveforms are available in tens or hundreds of microseconds, allowing for quick response after abrupt changes in the desired torque. We demonstrate our approach on a simple numerical example, in which we give the optimal waveforms for a motor with a sinusoidal back-EMF, and for a motor with a more complicated, nonsinusoidal waveform, in both the constant-torque region and constant-power region.

Keywords: brushless DC motors; permanent magnet synchronous motors; optimal control; convex optimization; ADMM

1. Introduction

We consider the problem of controlling an AC permanent magnet synchronous motor (PMSM) by choosing phase winding current waveforms that produce smooth output torque. Traditionally, the problem is solved differently depending on the type of motor: if the rotor magnets induce an counter-electromotive force (back-EMF) in the phase windings that is a sinusoidal function of rotor position (i.e., the motor has a *sinusoidal back-EMF waveform*), then sinusoidal current waveforms are used; if the induced back-EMF is instead a trapezoidal function (i.e., the motor has a *trapezoidal back-EMF waveform*), then rectangular current waveforms are used. Both of these schemes produce smooth output torque. For general back-EMF waveforms, however, there may not exist simple formulas for expressing current waveforms that produce smooth torque, especially when other constraints, such as voltage limits, are taken into account. This paper gives a numerical method for generating such waveforms.

In particular, we address the problem of choosing drive current waveforms that achieve a desired average torque while minimising a combination of resistive power loss and root-mean-square (RMS) torque ripple. We consider motors with general back-EMF waveforms. We assume supply voltage limits and phase current limits due to magnetic saturation. We also include a simple eddy current loss model, which has the effect of penalising high-frequency harmonics of the current waveforms. Because our formulation can be applied to motors with arbitrary phase connections (including delta, wye, and independently connected phases), we can handle several fault conditions, (in 6 Section 6, we demonstrate operation of a delta-wound, three-phase motor with a single open-phase fault). We also discuss simple variations of our formulation, including alternative definitions of torque ripple (e.g., as the range of the torque values, or the mean absolute deviation), or maximum torque problems. We show that the proposed torque control problem, and all proposed variations, are convex optimisation problems, and can therefore be quickly and reliably solved using convex optimisation. We show how to use the alternating direction method of multipliers (ADMM) to solve the resulting optimisation problem, and we demonstrate that the algorithm can be executed quickly, typically well under one millisecond. Furthermore, the iterates of ADMM provide a very good approximation of the optimal waveforms, even before convergence. Indeed, the first iteration of the algorithm produces the optimal current waveforms when voltage and current limits are ignored, and can typically be computed in tens of microseconds, which is competitive with evaluating some of the analytical expressions given in the literature (see below). Within a few tens of iterations, the iterates are typically within a few tenths of a per cent of

* Corresponding author. Email: moehle@stanford.edu

the optimal values, and can be considered converged for all practical purposes. The algorithm can also incorporate new desired torque signals after each iteration, and new iterates can be implemented immediately, which reduces the torque response time to tens of microseconds on a standard processor, or hundreds of microseconds on a low-cost ARM processor, which is around the switching period of common power electronic devices. The inverter bridge voltage waveforms that generate the optimal current waveforms are also computed as a by-product of computing the optimal current waveforms; these open-loop optimal bridge voltages and current waveforms can be used in a closed-loop current control scheme.

The ability to find the optimal current waveforms in real time allows us to change the model parameters on the fly (e.g., changing phase resistance with temperature, or updating the inverter bus voltage), or change the problem based on operating requirements. For example, an electric vehicle application may require maximising output torque at some times, thus increasing the performance of the vehicle, and high efficiency and low torque ripple at other times, thus increasing the efficiency (and range) of the vehicle.

We give some numerical results for our method when applied to a simple motor model. In particular, we give the optimal waveforms for two motors, one with a sinusoidal back-EMF, and one with a (nearly) trapezoidal back-EMF, both in the constant-torque and constant-power regions. (The optimal drive currents are nonsinusoidal in the constant-power region, even for motors with sinusoidal back-EMF.) We additionally compare the performance of sinusoidal and trapezoidal back-EMF waveforms in this region, and we find that, when driven with the optimal current waveforms, the two types of motors perform similarly.

1.1 Related work

Convex optimisation:convex optimisation problems can be solved efficiently and reliably using standard techniques (Boyd & Vandenberghe, 2004). Recently, much work has been devoted to solving moderately sized convex optimisation problems quickly (i.e., in milliseconds or microseconds), possibly on embedded platforms, which enables convex-optimisation-based control policies to be implemented at kilohertz rates (O'Donoghue, Stathopoulos, & Boyd, 2013; Wang & Boyd, 2010). In addition, recent advances in automatic code generation for convex optimisation (Chu, Parikh, Domahidi, & Boyd, 2013; Mattingley & Boyd, 2010) can significantly reduce the cost and complexity of developing and verifying an embedded solver.

We provide an algorithm based on ADMM. Details about ADMM can be found in Boyd, Parikh, Chu, Peleato, and Eckstein (2011), Parikh and Boyd (2014). ADMM is is particularly well suited for real-time optimal control because it typically converges to acceptable accuracy very quickly, and because its simplicity allows for easily verifiable source code. Details about using ADMM for optimal control can be found in O'Donoghue et al. (2013).

PMSM current waveform optimisation:significant work has been devoted to finding optimal current waveforms for motors with nonsinusoidal back-EMF waveform, and several special cases have been solved, some analytically. Simple characterisations of current waveforms that minimise power loss and produce smooth torque in the absence of other constraints are given in Le-Huy, Perret, and Feuillet (1986), Hung and Ding (1992), Wu and Chapman (2005). The authors of Chapman, Sudhoff, and Whitcomb (1999) extend these results to include a trade-off between RMS torque ripple and power loss by solving a linear least squares problem. An RMS voltage limit is considered by Hanselman (1994), who proposes offline solution of the resulting quadratic program and implementation as a lookup table. The authors of Aghili, Buehler, and Hollerbach (2001, 2003) instead introduce a current saturation limit, and analytically solve the Karush–Kuhn–Tucker (KKT) conditions to find the optimal current waveforms that produce smooth torque. Several of the aforementioned results can be extended to apply if one or more phases are in an open fault condition; explicit derivation of the optimal waveforms in this case can be found in Baudart, Matagne, Dehez, and Labrique (2013) and references therein (which neglect voltage and current limits). The authors of Yang, Wang, Wu, and Luh (2004) take a different approach, showing that minimum power loss required to achieve a desired torque, without regard for torque ripple, is attained when the current waveform is proportional to the back-EMF waveform.

Field-oriented control:for a motor with sinusoidal back-EMF waveforms and no voltage limits, sinusoidal current waveforms minimise power loss while also producing no torque ripple. In this case, it is convenient to formulate the control problem using Park's transformation into the d–q reference frame, which separates the current producing component of the current waveform from the field-weakening component, resulting in a *field-oriented control* framework (Gabriel, Leonhard, & Nordby, 1980; Hendershot & Miller, 1994),

Within this framework, two general techniques can be used to minimise power loss in steady-state. In *loss-model control*, a model of the power loss is used to determine the sinusoidal currents that minimise power loss in steady-state at a given operating point. Early work in this area focused on finding analytical expressions for the optimal current for simple loss models (Chau, Chan, & Liu, 2008; Morimoto, Tong, Takeda, & Hirasa, 1994). More recent approaches have addressed nonlinearities due to magnetic saturation, by storing a lookup table of optimal d- and q-axis current components (Lee, Nam, Choi, & Kwon, 2007), or computing them in real time using Newton's method (Jeong, Sul, Hiti, and Rahman, 2006). *Search control* techniques instead involve actively searching for the optimal current

d- and q-axis current components by directly iterating on the motor itself (Colby & Novotny, 1988; Vaez, John, & Rahman, 1999).

Model predictive control:as an alternative to seeking the optimal steady-state currents, model predictive control has recently been proposed to improve dynamic response. Most formulations involve solving a convex optimisation problem to determine the inverter voltage signal, which can either be done offline and implemented as a lookup table, as in Wang, Chai, Yoo, Gan, and Ng (2014), Mariethoz, Domahidi, and Morari (2009), Bolognani, Kennel, Kuehl, and Paccagnella (2011), Bolognani, Bolognani, Peretti, and Zigliotto (2009) or online in real time, as in Stumper, Dötlinger, and Kennel (2012). A different model predictive control scheme is proposed in Geyer (2011), who considers the discrete switching states of the inverter. Using various heuristic strategies, they are able to compute a sequence of inverter switching states quickly enough for embedded implementation.

1.2 Outline

In Section 2, we introduce our model of the PMSM. In Section 3, we formally introduce the torque control problem, and we list several variations of the base problem. We then explore the types of symmetry that a PMSM typically exhibits, and we show how to use symmetry to reduce the complexity of the torque control problem. In Section 4, we demonstrate the solution of the torque control problem, using ADMM. In Section 5, we discuss how to implement the solution in real time, possibly on an embedded processor. In Section 6, we show the optimal waveforms for an example motor, which we use to compare rectangular and sinusoidal back-EMF waveforms, and to compare the optimal waveforms with sinusoidal current waveforms.

2. Model

The model describes a three-phase PMSM, shown in Figure 1. The rotor, which is nonconductive and contains permanent magnets, has angular position θ and angular velocity ω; we assume ω is constant. The stator contains three circuits, called *phase windings*, with currents i_a, i_b, and i_c, and voltages v_a, v_b, and v_c. The phase currents induce eddy currents in the stator iron; we consider induced eddy currents separately for each phase, which we call j_a, j_b, and j_c. The motor is driven by a voltage-source, triple-half-bridge inverter with bridge voltages v_U, v_V, and v_W. The output torque is τ.

We will assume that i_a, i_b, i_c, v_a, v_b, v_c, j_a, j_b, j_c, v_U, v_V, v_W, and τ are 2π-periodic functions of θ. We use a prime ($'$) to denote differentiation of these functions with respect to θ. To lighten notation, we often drop explicit dependence on θ.

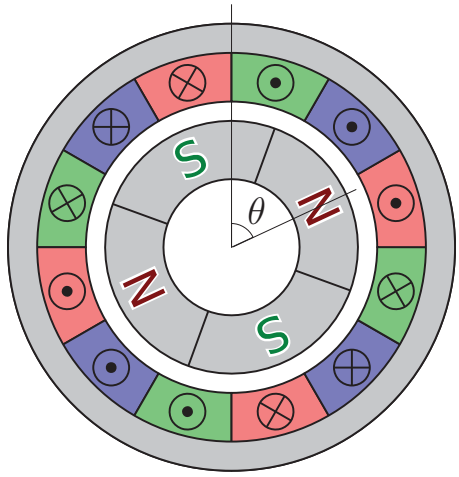

Figure 1. Schematic of permanent magnet synchronous motor. The rotor has angular position θ. Symbols $\otimes$ and $\odot$ represent the direction of axial windings. Different phase windings are in different shades.

2.1 Dynamics

The circuit dynamics of the phase windings are

$$\begin{aligned} v_a &= Ri_a + \omega(Li_a' + Mi_b' + Mi_c' + \tilde{M}j_a' + k_a), \\ v_b &= Ri_b + \omega(Mi_a' + Li_b' + Mi_c' + \tilde{M}j_b' + k_b), \\ v_c &= Ri_c + \omega(Mi_a' + Mi_b' + Li_c' + \tilde{M}j_c' + k_c), \end{aligned} \tag{1}$$

where R is the phase resistance, L is the phase self inductance, M is the mutual inductance between phases, and $\tilde{M}$ is the mutual inductance between a phase and the corresponding eddy current. The back-EMF waveforms k_a, k_b, and k_c are 2π-periodic functions of θ. The eddy current circuits are modelled as RL (resistance–inductance) circuits, with dynamics

$$\begin{aligned} 0 &= \tilde{R}j_a + \omega(\tilde{L}j_a' + \tilde{M}i_a'), \\ 0 &= \tilde{R}j_b + \omega(\tilde{L}j_b' + \tilde{M}i_b'), \\ 0 &= \tilde{R}j_c + \omega(\tilde{L}j_c' + \tilde{M}i_c'), \end{aligned} \tag{2}$$

where $\tilde{R}$ and $\tilde{L}$ are respectively the eddy circuit resistance and self inductance.

2.2 Inverter

The inverter bridge voltages satisfy

$$|v_U| \le (1/2)V_{dc}, \quad |v_V| \le (1/2)V_{dc}, \quad |v_W| \le (1/2)V_{dc}, \tag{3}$$

where V_{dc} is the constant DC bus voltage (Figure 2). The relation between the bridge voltages and the phase voltages depends on the winding connection, and is given below for three common winding connections.

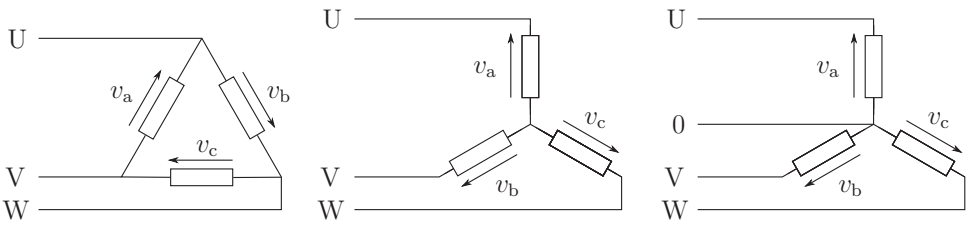

Figure 2. Delta, wye, and independently driven phase connections.

- Delta connection:

$$v_a = v_U - v_V, \quad v_b = v_V - v_W, \quad v_c = v_W - v_U. \tag{4}$$

- Wye connection:

$$\begin{aligned} &v_a - v_b = v_U - v_V, \quad v_b - v_c = v_V - v_W, \\ &v_c - v_a = v_W - v_U, \end{aligned} \tag{5}$$

  as well as Kirchoff's current law for the centre node:

$$i_a + i_b + i_c = 0. \tag{6}$$

- Independent phases:

$$v_a = v_U, \qquad v_b = v_V, \qquad v_c = v_W. \tag{7}$$

Other configurations are possible. For example, a six-half-bridge inverter can arbitrarily assign voltages to each side of the three phases (with each bridge voltage between $-(1/2)V_{dc}$ and $(1/2)V_{dc}$).

2.3 Magnetic saturation

We assume that the phase currents are maintained within the following magnetic saturation limits:

$$|i_a| \le i^{max}, \qquad |i_b| \le i^{max}, \qquad |i_c| \le i^{max}. \tag{8}$$

2.4 Torque

The total output torque is

$$\tau = k_a i_a + k_b i_b + k_c i_c + \tau_{cog}, \tag{9}$$

where the cogging torque τ_{cog} is a 2π-periodic function of θ. The average torque over one cycle is

$$\overline{\tau} = \frac{1}{2\pi} \int_0^{2\pi} \tau \, d\theta.$$

The RMS torque ripple is

$$r_{RMS} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} (\tau - \overline{\tau})^2 \, d\theta},$$

and the relative torque ripple is

$$r_{RMS,rel} = \frac{r_{RMS}}{\overline{\tau}}.$$

2.5 Power loss

The power loss is the average resistive loss from all phase currents and eddy currents over one cycle:

$$P_{loss} = \frac{1}{2\pi} \int_0^{2\pi} \left(R\left(i_a^2 + i_b^2 + i_c^2\right) + \tilde{R}\left(j_a^2 + j_b^2 + j_c^2\right) \right) d\theta.$$

The relative power loss is

$$P_{loss,rel} = \frac{P_{loss}}{\overline{\tau}\omega},$$

and the efficiency is $\eta = 1 - P_{loss,\,rel}$. This assumes that $\overline{\tau}\omega > 0$, i.e., the mechanical output power is positive. When the motor is used as a generator, or for regenerative braking, the denominator can be replaced by its absolute value.

3. Torque control problem

3.1 Optimal torque control

The optimal torque control problem is to choose the phase voltages, phase currents, and eddy currents to achieve a desired average torque while minimising the average power loss and torque ripple:

$$\begin{array}{ll} \text{minimise} & P_{loss} + \lambda r_{RMS}^2 \\ \text{subject to} & \overline{\tau} = \tau^{des}, \\ & \text{Equations (1), (2), (3), (8), (9),} \\ & \text{and one of (7), (4), or (5)–(6).} \end{array} \tag{10}$$

The parameters are the trade-off parameter $\lambda \ge 0$, the circuit parameters R, L, and M, the eddy circuit parameters $\tilde{R}$, $\tilde{L}$, and $\tilde{M}$, the current and voltage limits i^{max} and V_{dc}, the shaft

speed ω, the desired average torque τ^{des}, and the waveforms k_a, k_b, and k_c. The variables are the 2π-periodic functions $i_a, i_b, i_c, j_a, j_b, j_c, v_a, v_b, v_c, v_U, v_V, v_W$, and τ. The constraints include the dynamics, current, and voltage limits, torque–current relation, and one set of winding constraints.

Problem (10) is an infinite-dimensional (convex) quadratic program, since the variables to be determined are (periodic) functions. Once we discretise the variables, it can be (approximately) solved quickly and reliably using standard methods for convex optimisation (e.g., interior point; see Boyd and Vandenberghe (2004) for details).

3.2 Variations

We list some variations of (10) that also result in convex optimisation problems.

Power loss and torque ripple constraints:we can limit the acceptable power loss and torque ripple by adding the constraints

$$P_{\text{loss}} \le P_{\text{loss}}^{\max}, \qquad r_{\text{RMS}} \le r_{\text{RMS}}^{\max}. \tag{11}$$

Alternatively, we can use the relative values of power loss or relative torque ripple in the objective or in (11).

Maximum torque: we can set up a maximum torque problem. To do this, we remove the $\overline{\tau} = \tau^{\text{des}}$ constraint. Then instead of minimising the power loss and torque ripple, we maximise $\overline{\tau}$. Other constraints may be added; for example, a power loss constraint can be used to obtain the maximum sustainable torque subject to power loss limits.

Alternative ripple definitions: we can use other definitions for the torque ripple, such as

$$r_{\text{range}} = \sup_{\theta} \tau(\theta) - \inf_{\theta} \tau(\theta)$$

or

$$r_{\text{abs}} = \frac{1}{2\pi} \int_0^{2\pi} |\tau - \overline{\tau}| \, d\theta.$$

These are both convex, nonquadratic functionals of τ.

Mitigation of harmonics: we can establish magnitude constraints or penalties on the magnitudes of specified harmonics of the currents or torque (e.g., to avoid a known mechanical resonance).

Open-phase fault: we can continue operation with one inoperable phase winding by removing the relevant equations and variables from the model.

3.3 Symmetry

Although the variables and parameters of (10) are fully defined by their values over the interval $[0, 2\pi]$, we can use the following assumptions to make this interval shorter, thus reducing the complexity of a discretised version of (10). This exploitation of symmetry does not change the solution; it merely results in a smaller discretised problem that can be solved more quickly. For some asymmetric motors, these assumptions may not hold; examples of this include motors intentionally designed without symmetry, or when a winding in an otherwise symmetric motor has failed.

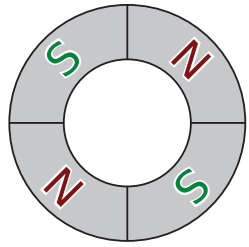

Figure 3. A rotor with pole symmetry ($N_p = 2$). The magnetic field generated by the permanent magnets is identical at θ and $\theta + \pi$.

Pole symmetry: we assume the rotor has N_p pole pairs (Figure 3), i.e., k_a, k_b, k_c, and τ_{cog} are $2\pi/N_p$-periodic. Consequently, there is a solution of (10) in which the optimal variables are also $2\pi/N_p$-periodic.

Half-wave symmetry:we further assume that k_a, k_b, k_c and τ_{cog} are half-wave symmetric with period $2\pi/N_p$ (e.g., $k_a(\theta) = -k_a(\theta + \pi/N_p)$). This implies there exists a solution of (10) in which the optimal variables are also half-wave symmetric with period $2\pi/N_p$.

Phase symmetry: we assume the stator windings are displaced by $2\pi/(3N_p)$ radians from each other, so the back-EMF waveform in the second and third phases are shifted versions of the first:

$$k_b(\theta) = k_a\left(\theta + \frac{2\pi}{3N_p}\right), \quad k_c(\theta) = k_a\left(\theta - \frac{2\pi}{3N_p}\right) \tag{12}$$

and τ_{cog} is $2\pi/(3N_p)$-periodic. This implies there exists a solution of the original problem in which the optimal variables are shifted versions of each other:

$$i_b(\theta) = i_a\left(\theta + \frac{2\pi}{3N_p}\right), \quad i_c(\theta) = i_a\left(\theta - \frac{2\pi}{3N_p}\right), \tag{13}$$

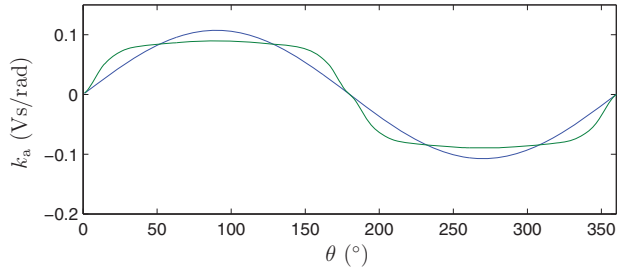

Figure 4. Sinusoidal and trapezoidal back-EMF waveform for the numerical example.

with similar shift relations for the voltage and eddy current. These assumptions, combined with (9) and (13), imply τ is $2\pi/(3N_\mathrm{p})$-periodic.

Equivalent problem: the symmetry assumptions allow us to form an equivalent problem with the same constraints and objective as (10) in which the variables have domain $[0, \pi/(3N_\mathrm{p})]$. We also add periodicity constraints of the form

$$i_\mathrm{a}(0) = -i_\mathrm{c}\left(\frac{\pi}{3N_\mathrm{p}}\right), \quad i_\mathrm{b}(0) = -i_\mathrm{a}\left(\frac{\pi}{3N_\mathrm{p}}\right),$$
$$i_\mathrm{c}(0) = -i_\mathrm{b}\left(\frac{\pi}{3N_\mathrm{p}}\right),$$

with similar constraints for the voltage and eddy currents, and $\tau(0) = \tau(\pi/(3N_\mathrm{p}))$. The integrands in the definitions of average torque, torque ripple, and power loss are each $\pi/(3N_\mathrm{p})$-periodic; to get equivalent definitions of these values over the appropriate domain, we can integrate over $[0, \pi/(3N_\mathrm{p})]$ instead of $[0, 2\pi]$, and scale the result by $6N_\mathrm{p}$.

Any set of optimal variables for the reduced problem is the restriction to $[0, \pi/(3N_\mathrm{p})]$ of some set of optimal variables to the original problem. To reconstruct the values of these variables over the rest of the interval $[0, 2\pi]$, we can use the $2\pi/N_\mathrm{p}$-periodicity and half-wave symmetry of the variables, as well as the phase symmetry shift relations (such as (13)).

4. Solution

4.1 Discretisation

After reducing the domain of the variables of (10), we discretise this interval into $N + 1$ grid points, $\theta_0, \ldots, \theta_N$, with $\theta_0 = 0$ and $\theta_N = \pi/(3N_\mathrm{p})$. All pointwise constraints must hold at $\theta_0, \ldots, \theta_{N-1}$, and the periodicity constraints must hold at θ_0 and θ_N. Integration over the interval is replaced by summation from θ_0 to θ_{N-1}, with appropriate scaling.

Because the dynamics equations (1) and (2) are linear, they can be discretised using an exact method, such as first-order hold; however, for an embedded application, a faster (but less accurate) discretisation method, such as forward Euler, may be more practical. Based on our (informal) observations, this inaccuracy does not appear to significantly degrade the quality of the solution.

Finite-dimensional quadratic program:after discretisation, problem (10) can be expressed as

$$\begin{array}{ll} \text{minimise} & x^T P x + q^T x + r \\ \text{subject to} & Ax = b, \\ & |x_i| \le c_i, \quad i = 1 \ldots n. \end{array} \tag{14}$$

The variable $x \in \mathbf{R}^n$ is a vector containing the variables of (10) at $\theta_0 \ldots \theta_N$, and the parameters are the symmetric positive-definite matrix $P \in \mathbf{R}^{n\times n}$, as well as $q \in \mathbf{R}^n$, $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$, $r \in \mathbf{R}$, and $c_i \in \mathbf{R} \cup \{\infty\}$. We note that the parameter matrices P and A are sparse, which can be exploited in a numerical solution.

4.2 Interior-point methods

Problem (14) can be solved using a generic interior-point solver, which involves solving a sequence of linear systems of equations (around 15) of size $(m + n)$. Using a generic quadratic programming solver, the size of these linear systems scales by the cube of the number of grid points. However, a method that exploits the sparse structure of the problem can reduce this to be proportional to the number of grid points. Several software packages are available that can do this, and are capable of running on embedded platforms (Chu et al., 2013; Mattingley & Boyd, 2012; Wang & Boyd, 2010).

4.3 ADMM

In addition to interior-point methods, we propose to solve (14) using ADMM. Starting from any $x^{(0)}$, $z^{(0)}$, and $y^{(0)}$, the algorithm generates iterates $x^{(k+1)}$, for $k = 0, 1, \ldots$, according to

$$x^{(k+1)} = \operatorname*{argmin}_x \left(f(x) + (1/2)\rho\|x - y^{(k)}\|^2\right) \tag{15}$$
$$z^{(k+1)} = \mathrm{sat}\left(2x^{(k+1)} - y^{(k)}, c\right) \tag{16}$$
$$y^{(k+1)} = y^{(k)} + z^{(k+1)} - x^{(k+1)}, \tag{17}$$

where $\|\cdot\|$ denotes the ℓ_2 norm, $\rho > 0$ is an algorithm parameter,

$$f(x) = \begin{cases} x^T P x + q^T x & Ax = b \\ \infty & \text{otherwise,} \end{cases}$$

and sat is the vector saturation function, i.e., the ith element of $sat(x, c)$ is

$$\left(\mathrm{sat}(x, c)\right)_i = \begin{cases} -c_i & x_i < -c_i \\ c_i & x_i > c_i \\ x_i & \text{otherwise.} \end{cases} \tag{18}$$

Note that $x^{(k+1)}$ can be interpreted as a solution of a regularised version of (14) with no inequality constraints (i.e., $c_i = \infty$), and $z^{(k+1)}$ as a projection onto the set of points satisfying the inequality constraints of (14).

Finding $x^{(k+1)}$ that satisfies (15) can be done by solving

$$\begin{bmatrix} P + \rho I & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} x^{(k+1)} \\ \mu \end{bmatrix} = \begin{bmatrix} y^{(k)} - q \\ b \end{bmatrix}, \tag{19}$$

Table 1. Average solve times in milliseconds.

	Intel Xeon	ARM processor
CVX	468.67	–
CVXGEN	3.9	229.8
ADMM (cold start)	0.12	3.94
ADMM (warm start)	0.07	2.71

for $x^{(k+1)}$ and μ. Because each iteration of ADMM involves solving (19) for a different value of $z^{(k)}$, it is convenient to store the coefficient matrix (the KKT matrix) in factored form using a sparse LDL factorisation.

Convergence of $x^{(k)}$ to a solution of (14) is guaranteed (Boyd et al., 2011). In practice, however, it may be beneficial to iterate over (15)–(17) continuously, updating the (factorised) KKT matrix periodically to reflect the new values of the parameters (in particular, ω). This can be interpreted as a *warm start* for a new instance of (14); because the entries of the KKT matrix change little when updated, we expect the iterates of the previous problem instance to be nearly optimal for the new problem. We therefore can always implement the waveforms contained in the most recent current iterates immediately, because they reflect the current values of all parameters and we expect that they are always very close to optimality. (Indeed, even the first iterate $x^{(0)}$ is the solution of a regularised version of (14) without voltage or current limits.) In addition, because only q and b in (19) depend on τ^{des}, we can update the desired torque after every iteration without refactoring the KKT matrix. Preliminary waveforms that reflect changes in τ^{des} can therefore be implemented after a single linear system solve.

4.4 Solve times

We compare the average solve times of ADMM against two interior-point solvers using the parameter values of the example in Section 6 (Table 2). We used $N = 15$ (i.e., 90 grid points per cycle).

All three algorithms were carried out on a Linux machine with a 3.4 GHz Intel Xeon E31270 processor. In addition, CVXGEN and ADMM were carried out a Raspberry Pi, a $25 computer with a 700 Mhz ARM processor with a floating point unit. Table 1 gives the average solve times for all three algorithms over 1000 uniformly randomly selected torque-speed pairs in the feasible operating region of the motor.

CVX: CVX (Grant & Boyd, 2008, 2013) is a MATLAB-based modelling language for convex optimisation. It converts (14) into a second-order cone program, which is solved using the interior-point solver SDPT3 (Toh, Todd, & Tütüncü, 1999; Tütüncü, Toh, & Todd, 2003).

CVXGEN: CVXGEN (Mattingley & Boyd, 2012) is a code generator for fast convex optimisation. It takes a high-level description of a quadratic program and generates a custom interior-point solver, written in C, which is suitable for embedded, real-time optimisation. For details on code generation for real-time convex optimal control, see Mattingley, Wang, and Boyd (2011), Mattingley and Boyd (2010). CVXGEN was terminated once the duality gap was less than 0.1.

ADMM: ADMM was implemented in C. The variables v_a, v_b, and v_c were (analytically) eliminated, resulting in a problem with 150 variables, 107 equality constraints, and 90 inequality constraints. Equation (19) was solved using the LDL package provided in Davis (2005). The algorithm was terminated once $\|x^{(k)} - z^{(k)}\|_\infty$ was less than 0.1.

If a sequence of similar problem instances is to be solved, we can accelerate convergence of each iteration by initialising the iterates at the solution to the previous problem (called *warm starting*). We provide solve times for ADMM for both cold starting (iterates initialised to zero) and warm starting (iterates initialised to the solution of an identical problem, but with ω uniformly randomly perturbed to a feasible value within 20%).

5. Implementation

Here we collect several ideas for implementing the solution to (10).

Table lookup: any of the above solvers can be used to to generate a lookup table of optimal waveforms indexed by ω and τ^{des}. This enables the optimal current waveforms to be used on processors with limited computational capability. Some simplifications may be helpful to reduce storage requirements; for example, for motors with wye or independent phase connections, neglecting eddy current, a single current wave shape can be stored for any ω and τ^{des} such that the inequality constraints of (10) are inactive. This wave shape can be scaled to meet the torque requirement.

Real-time optimisation. our implementations of CVXGEN and ADMM were fast enough to compute the optimal waveforms on an embedded system. This allows us to recompute the optimal waveforms after updating the model parameters (e.g., after updating winding resistance with temperature) or after changing the trade-off parameter λ based on performance requirements. We can also change the problem entirely to one of the variations mentioned in Section 3.2, such as a maximum-torque mode, or continued operation after failure of a phase winding.

Feedback control: if the model is perfectly correct, using the optimal bridge voltage waveforms as PWM signals for the inverter will produce the resulting optimal current waveforms. In practice, a closed-loop controller is necessary to ensure accurate tracking of the optimal waveforms For example, a simple state-feedback controller would have the

form

$$\begin{bmatrix} v_{\mathrm{U}} \\ v_{\mathrm{V}} \\ v_{\mathrm{W}} \end{bmatrix} = \begin{bmatrix} v_{\mathrm{U}}^{\star} \\ v_{\mathrm{V}}^{\star} \\ v_{\mathrm{W}}^{\star} \end{bmatrix} + K \left(\begin{bmatrix} i_{\mathrm{a}} \\ i_{\mathrm{b}} \\ i_{\mathrm{c}} \end{bmatrix} - \begin{bmatrix} i_{\mathrm{a}}^{\star} \\ i_{\mathrm{b}}^{\star} \\ i_{\mathrm{c}}^{\star} \end{bmatrix} \right), \qquad (20)$$

where we use the $\star$ to denote the optimal (reference) values, and $K \in \mathbf{R}^{3\times 3}$ is a controller synthesised based on the circuit dynamics (1). Note that the inclusion of the optimal open-loop bridge voltages will improve dynamic performance compared to a simple tracking controller. Extension to more complex controller architectures (e.g., PI controllers) is beyond the scope of this paper.

6. Example

In this section, we demonstrate the optimal waveforms for a numerical example. The values of all scalar model parameters are given in Table 2. The phase resistance, phase inductance, and voltage limits are based on the first example of Liu, Zhu, and Howe (2005). The eddy circuit parameters were adjusted heuristically to produce reasonably smooth waveforms. Because the effects of the eddy circuit dynamics are fully described using only two parameters, we arbitrarily set $\tilde{M} = 1$ mH. Cogging torque is neglected. The motor described in Liu et al. (2005) has a sinusoidal back-EMF waveform with RMS value of 0.72 V/s. For comparison, we will also consider an empirically determined trapezoidal

Table 2. Motor parameters.

Parameter	Value	Unit
R	0.466	Ω
L	3.19	mH
M	−1.31	mH
$\tilde{R}$	4.6	Ω
$\tilde{L}$	1.1	mH
$\tilde{M}$	1.0	mH
V_{dc}	70	V
$i^{\max}$	10	A
N_p	1	–

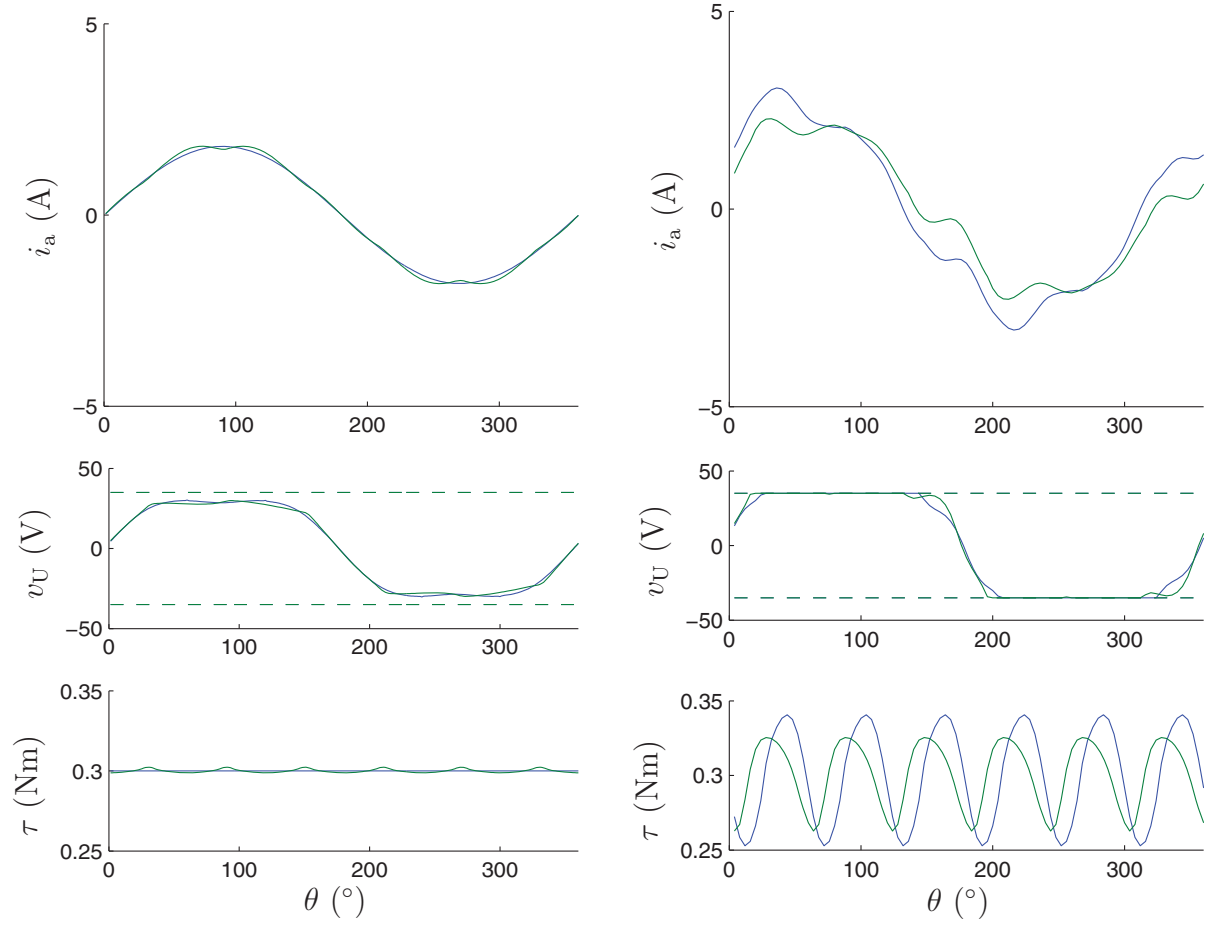

Figure 5. (Colour online) The optimal current and voltage waveforms for the both sinusoidal and trapezoidal back-EMF waveforms, for $\tau^{\mathrm{des}} = 0.3$ Nm and $\lambda = 2$ kW/(Nm)^2. The left and right figures show $\omega = 300$ rad/s and $\omega = 400$ rad/s, respectively, and the blue and green lines correspond to the sinusoidal and trapezoidal back-EMF waveforms, respectively.

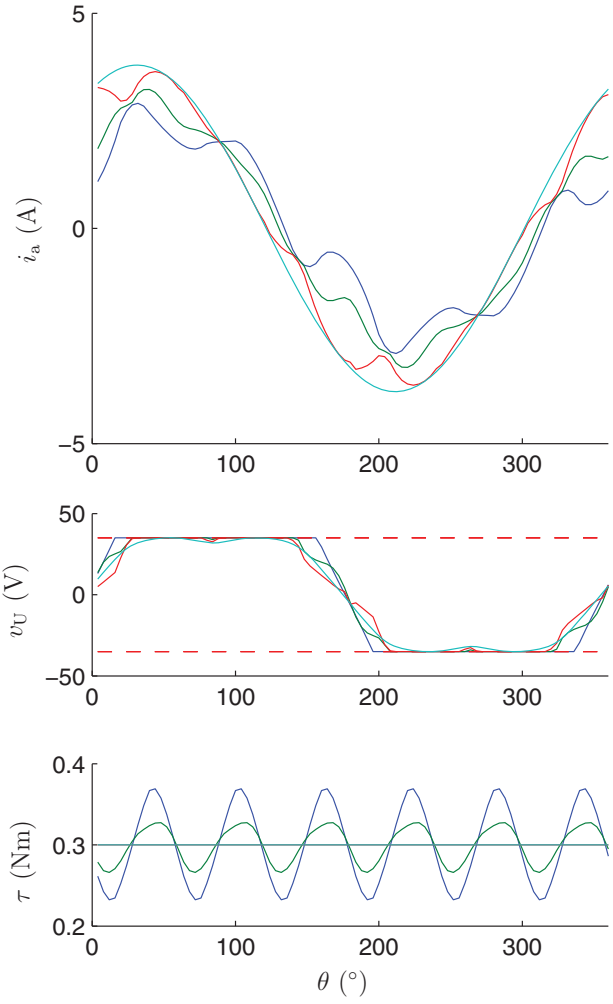

Figure 6. (Colour online) The optimal waveforms for the sinusoidal back-EMF, for $\omega = 400$ rad/s, and $\tau^{\text{des}} = 0.3$ Nm, for three values of λ, with the optimal sinusoidal currents shown for comparison. The blue, green, and red lines give correspond to $\lambda = 0$ kW/(Nm)^2, $\lambda = 2$ kW/(Nm)^2, and $\lambda \to \infty$, respectively, and the cyan line corresponds to the sinusoidal current.

back-EMF waveform (obtained from Park, Park, Lee, and Harashima (2000)) with the same RMS value as the sinusoidal back-EMF waveform. Both back-EMF waveforms are shown in Figure 4.

6.1 *Sinusoidal vs. rectangular back-EMF*

We demonstrate the solution of (10) for both the sinusoidal and the trapezoidal back-EMF waveforms for $\tau^{\text{des}} = 0.3$ Nm. In Figure 5, the optimal waveforms for $\omega = 300$ rad/s, which is below the rated speed of the motor, are shown in Figure 5. We first note that the optimal current waveforms for the sinusoidal back-EMF are sinusoidal, verifying the classical result that sinusoidal currents simultaneously minimise power loss and achieve smooth torque.

The optimal waveforms for the trapezoidal motor are more subtle. Because the constraints are inactive, (10) can

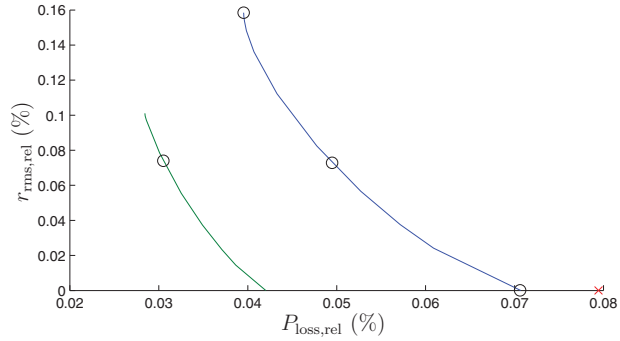

Figure 7. (Colour online) The achievable relative power loss and relative torque ripple for $\omega = 400$ rad/s and $\tau^{\text{des}} = 0.3$ Nm, for both back-EMF waveforms (sinusoidal in blue and trapezoidal in green). The circles correspond to the waveforms shown in Figures 5 and 6, and the $\times$ corresponds to the optimal sinusoidal waveforms.

be solved in exactly one iteration of ADMM with ρ set to zero. Also, if eddy current is neglected, the optimal current waveforms are independent of ω. (This is not the case for delta-connected motors, in which the back-EMF can induce circulating current which cannot be controlled by the inverter.)

We also compare the optimal waveforms at $\omega = 400$ rad/s, which is above rated speed of the motor. In this case, the optimal current waveforms are no longer sinusoidal, for either back-EMF waveform.

6.2 *Optimal vs. sinusoidal current waveforms*

In the last section, we saw that sinusoidal current waveforms are not optimal in the constant power region for all values of λ, even with a sinusoidal back-EMF waveform. Here, we show that, for $\omega = 400$ rad/s and $\tau^{\text{des}} = 0.3$ Nm, there is no value of λ for which sinusoidal current waveforms are optimal. Figure 6 shows the optimal waveforms for $\lambda = 0$ kW/(Nm)^2, $\lambda = 2$ kW/(Nm)^2, and as $\lambda \to \infty$. As expected, we find that larger values of λ result in lower torque ripple, but require greater phase advance and a greater amplitude current waveform, resulting in greater power loss. We also show the optimal sinusoidal currents for comparison, which also can be found using convex optimisation, by solving a special case of the harmonic mitigation problem of Section 3.2 (because sinusoidal current waveforms always produce a smooth torque output, the choice of λ is irrelevant). Note that the sinusoidal current waveforms have noticeably higher peak value than any of the other waveforms. When compared with the optimal waveform for $\lambda \to \infty$, the sinusoidal waveform has a greater magnitude for all θ, which immediately indicates higher power loss.

By varying λ, we can characterise the entire trade-off curve between power loss and torque ripple (shown in Figure 7). Any point on this curve is optimal for some positive value of λ, and the points corresponding to the waveforms

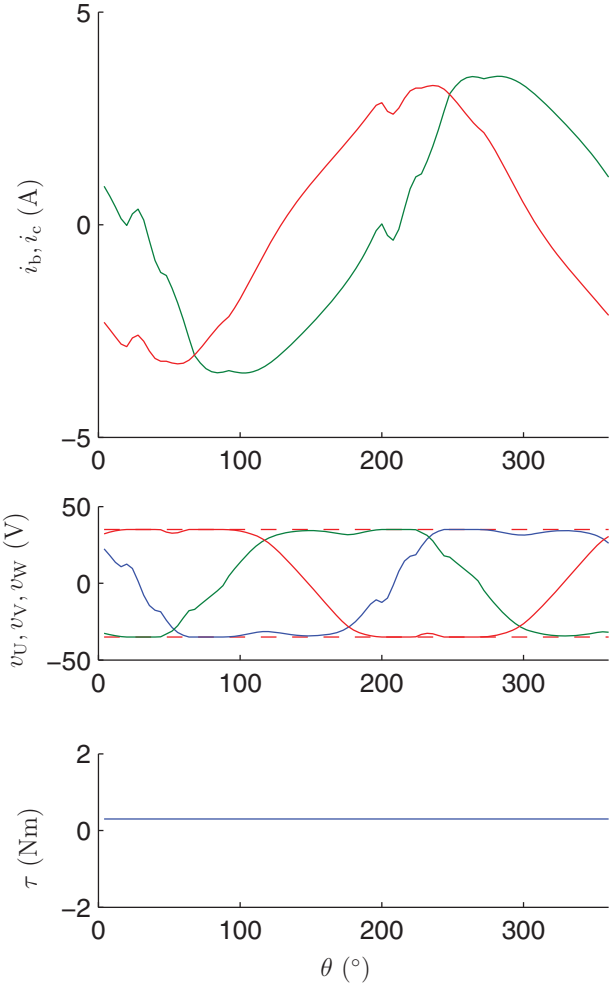

Figure 8. The minimum-ripple optimal current and voltage waveforms for $\omega = 650$ rad/s, $\tau^{\text{des}} = 0.3$ Nm), for a delta-connected motor with a single open-phase fault.

of Figures 5 and 6 are shown. The point corresponding to the optimal sinusoidal current waveforms does not lie on the curve, indicating that they are not optimal for any positive λ. Indeed, by using the optimal waveforms at this point, we can increase efficiency by several per cent, depending on our tolerance for torque ripple.

We also show the curve corresponding to the optimal waveforms for the trapezoidal back-EMF, which in this case strictly outperforms the sinusoidal back-EMF, assuming optimal waveforms are used.

6.3 Open-phase fault

We demonstrate the ability of a delta-connected motor to operate if one winding has failed in open circuit, and we find that this is possible, even with active voltage constraints. Note that in this case, only the first of the three symmetry assumptions (pole symmetry) holds. In Figure 8, we show the optimal waveforms for $\omega = 650$ rad/s and $\tau^{\text{des}} = 0.3$ Nm), taking $\lambda \to \infty$, thus generating smooth torque.

7. Conclusion

In this paper, we pose torque control of brushless permanent magnet motors as an optimal control problem. In this problem, we minimise a (user-defined) combination of power loss and torque ripple while achieving a desired average torque and respecting bridge voltage and phase current limits (arising from saturation). The resulting problem (or one of the proposed variations) is convex, and can therefore be solved quickly and reliably. We give an algorithm (ADMM) which is fast enough to be implemented in real time (0.4 μs/solve), possibly on embedded platforms, and we give some practical recommendations to ensure a fast response (tens of microseconds) to changes in the desired torque.

We conclude by noting that for many classes of AC motors, (e.g., induction machines and switched reluctance machines), a similar torque control problem would be a nonconvex optimisation problem (due to nonlinear dynamics or a nonlinear relation between torque and current). Nonconvex optimisation problems are hard to solve in general, and ADMM is not gauranteed to converge for these problems. We note, however, that for many nonconvex optimisation problems, ADMM appears to work well in practice (see Boyd et al. (2011) for details).

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Aghili, F., Buehler, M., & Hollerbach, J.M. (2001). Quadratic programming in control of brushless motors. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation* (vol. 2, pp. 1130–1135).

Aghili, F., Buehler, M., & Hollerbach, J.M. (2003). Experimental characterization and quadratic programming-based control of brushless-motors. *IEEE Transactions on Control Systems Technology, 11*(1), 139–146.

Baudart, F., Matagne, E., Dehez, B., & Labrique, F. (2013). Optimal current waveforms for torque control of permanent magnet synchronous machines with any number of phases in open circuit. *Mathematics and Computers in Simulation, 90*, 1–14.

Bolognani, S., Bolognani, S., Peretti, L., & Zigliotto, M. (2009). Design and implementation of model predictive control for electrical motor drives. *IEEE Transactions on Industrial Electronics, 56*(6), 1925–1936.

Bolognani, S., Kennel, R., Kuehl, S., & Paccagnella, G. (2011). Speed and current model predictive control of an IPM synchronous motor drive. In *IEEE International Electric Machines & Drives Conference* (pp. 1597–1602). IEEE.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning, 3*(1), 1–122.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Chapman, P.L., Sudhoff, S.D., & Whitcomb, C.A. (1999). Optimal current control strategies for surface-mounted permanent-magnet synchronous machine drives. *IEEE Transactions on Energy Conversion, 14*(4), 1043–1050.

Chau, K.T., Chan, C.C., & Liu, C. (2008). Overview of permanent-magnet brushless drives for electric and hybrid electric

vehicles. *IEEE Transactions on Industrial Electronics, 55*(6), 2246–2257.

Chu, E., Parikh, N., Domahidi, A., & Boyd, S. (2013). Code generation for embedded second-order cone programming. In *Proceedings of the 2013 European Control Conference* (pp. 1547–1552).

Colby, R.S., & Novotny, D.W. (1988). An efficiency-optimizing permanent-magnet synchronous motor drive. *IEEE Transactions on Industry Applications, 24*(3), 462–469.

Davis, T.A. (2005). Algorithm 849: A concise sparse Cholesky factorization package. *ACM Transactions on Mathematical Software, 31*(4), 587–591.

Gabriel, R., Leonhard, W., & Nordby, C.J. (1980). Field-oriented control of a standard AC motor using microprocessors. *IEEE Transactions on Industry Applications, 16*(2), 186–192.

Geyer, T. (2011). Computationally efficient model predictive direct torque control. *IEEE Transactions on Power Electronics, 26*(10), 2804–2816.

Grant, M., & Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, & H. Kimura (Eds.), *Recent advances in learning and control*, Lecture Notes in Control and Information Sciences (pp. 95–110). London: Springer-Verlag Limited.

Grant, M., & Boyd, S. (2013). CVX: Matlab software for disciplined convex programming [Online]. Retrieved from http://cvxr.com/cvx.

Hanselman, D. (1994). Minimum torque ripple, maximum efficiency excitation of brushless permanent magnet motors. *IEEE Transactions on Industrial Electronics, 41*(3), 292–300.

Hendershot, J.R., & Miller, T. (1994). *Design of brushless permanent-magnet machines*. Venice, FL: Motor Design Books.

Hung, J.Y., & Ding, Z. (1992). Minimization of torque ripple in permanent magnet motors: A closed form solution. In *Proceedings of the 18th IEEE Industrial Electronics Conference* (pp. 459–463). IEEE.

Jeong, Y., Sul, S., Hiti, S., & Rahman, K.M. (2006). Online minimum-copper-loss control of an interior permanent-magnet synchronous machine for automotive applications. *IEEE Transactions on Industry Applications, 42*(5), 1222–1229.

Le-Huy, H., Perret, R., & Feuillet, R. (1986). Minimization of torque ripple in brushless DC motor drives. *IEEE Transactions on Industry Applications, 22*(4), 748–755.

Lee, J., Nam, K., Choi, S., & Kwon, S. (2007). A lookup table based loss minimizing control for FCEV permanent magnet synchronous motors. In *IEEE Vehicle Power and Propulsion Conference* (pp. 175–179). IEEE.

Liu, Y., Zhu, Z.Q., & Howe, D. (2005). Direct torque control of brushless DC drives with reduced torque ripple. *IEEE Transactions on Industry Applications, 41*(2), 599–608.

Mariethoz, S., Domahidi, A., & Morari, M. (2009). Sensorless explicit model predictive control of permanent magnet synchronous motors. In *IEEE International Electric Machines and Drives Conference* (pp. 1250–1257). IEEE.

Mattingley, J., & Boyd, S. (2010). Automatic code generation for real-time convex optimization. In Y. Eldar & D. Palomar (Eds.), *Convex optimization in signal processing and communications* (pp. 1–41). Cambridge: Cambridge University Press.

Mattingley, J. & Boyd, S. (2012). CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering, 13*(1), 1–27.

Mattingley, J., Wang, Y., & Boyd, S. (2011). Receding horizon control: Automatic generation of high-speed solvers. *IEEE Control Systems Magazine, 31*(3), 52–65.

Morimoto, S., Tong, Y., Takeda, Y., & Hirasa, T. (1994). Loss minimization control of permanent magnet synchronous motor drives. *IEEE Transactions on Industrial Electronics, 41*(5), 511–517.

O'Donoghue, B., Stathopoulos, G., & Boyd, S. (2013). A splitting method for optimal control. *IEEE Transactions on Control Systems Technology, 21*(6), 2432–2442.

Parikh, N., & Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization, 1*(3), 123–231.

Park, S.J., Park, H.W., Lee, M.H., & Harashima, F. (2000). A new approach for minimum-torque-ripple maximum-efficiency control of BLDC motor. *IEEE Transactions on Industrial Electronics, 47*(1), 109–114.

Stumper, J., Dötlinger, A., & Kennel, R. (2012). Classical model predictive control of a permanent magnet synchronous motor. *European Power Electronics and Drives Journal, 22*(3), 24–31.

Toh, K., Todd, M.J., & Tütüncü, R.H. (1999). SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optimization Methods and Software, 11*(1–4), 545–581.

Tütüncü, R.H., Toh, K.C., & Todd, M.J. (2003). Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming, 95*(2), 189–217.

Vaez, S., John, V.I., & Rahman, M.A. (1999). An on-line loss minimization controller for interior permanent magnet motor drives. *IEEE Transactions on Energy Conversion, 14*(4), 1435–1440.

Wang, L., Chai, S., Yoo, D., Gan, L., & Ng, K. (2014). *PID and predictive control of electrical drives and power converters*. Singapore: John Wiley and Sons.

Wang, Y., & Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology, 18*(2), 267–278.

Wu, A.P., & Chapman, P.L. (2005). Simple expressions for optimal current waveforms for permanent-magnet synchronous machine drives. *IEEE Transactions on Energy Conversion, 20*(1), 151–157.

Yang, Y., Wang, J., Wu, S., & Luh, Y. (2004). Design and control of axial-flux brushless DC wheel motors for electric vehicles—Part II: Optimal current waveforms and performance test. *IEEE Transactions on Magnetics, 40*(4), 1883–1891.