Mechanized Verification of Complex-to-Vector Derivations in a Three-Phase AC Machine Model

Pablo O. Cayuela

Universidad Tecnológica Nacional, Facultad Regional Córdoba, Argentina

ACL2 Workshop 2026

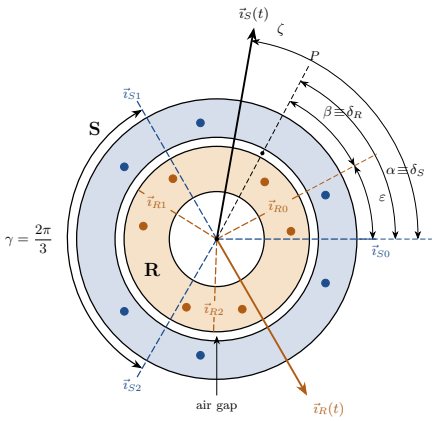
A sign in a published torque equation

- Engineering papers carry out long derivations by hand.
- The extended vectorial model of Leonhard's three-phase AC machine (CVM 2026, in Spanish) completes three steps that had been stated without proof:
  - the self and mutual inductances,
  - the three-phase collapse of the flux linkage,
  - the torque equation, from complex form to cross product.
- The last one gives a different sign from the published equation (eq. 46):

$$\vec{T}_M = \tfrac{2}{3} M \left[\vec{i}_S \times \vec{i}_R \, e^{+j\varepsilon}\right] \qquad \text{vs.} \qquad \tfrac{2}{3} M \left[\vec{i}_S \times \vec{i}_R \, e^{-j\varepsilon}\right] \text{ (original)}$$

- Saying a published equation is wrong needs more than a second read-through: every algebraic step checked by ACL2.

The machine and the vectorial model



Stator S and rotor R, windings $\gamma = 2\pi/3$ apart; rotor angle $\varepsilon(t)$.

Vectors in $\mathbb{R}^2$ = complex numbers; two identities do the work:

$$AB^* + A^*B = 2\,(\vec{A} \cdot \vec{B}) \qquad \text{(eq. 6–7)}$$

$$AB^* - A^*B = j2\,(\vec{A} \times \vec{B}) \qquad \text{(eq. 8–9)}$$

with $\vec{A} \times \vec{B} := \mathrm{Im}(AB^*)$ (Leonhard's form; the negative of Needham's).

Order of the derivation, followed by the books: currents and angle ζ_k (13)–(28) $\rightarrow$ mmf and induction (28a)–(29c) $\rightarrow$ inductances (29d)–(31) $\rightarrow$ flux linkage (32)–(35) $\rightarrow$ torque (36)–(46).

Full derivation, equations numbered as in the books: appendix (PDF) on GitHub, see last slide.

Representation: complex numbers as pairs

`complex-rationalp` needs rational parts; $\sqrt{3}/2$ and $\cos\theta$ are not.
So a complex number / planar vector is a `cons` of two ACL2 numbers:

```
(defun cplx (re im) (cons re im))
(defun cmul   (z w) (cplx (- (* (re z) (re w)) (* (imz z) (imz w)))
                          (+ (* (re z) (imz w)) (* (imz z) (re w)))))
(defun dotp   (z w) (+ (* (re z) (re w)) (* (imz z) (imz w))))
(defun crossp (z w) (- (* (imz z) (re w)) (* (re z) (imz w))))
```

With $A = a + jb$, $B = c + jd$, eqs. 6–9 are ring identities in four real components, no trigonometry:

$$AB^* + A^*B = 2(ac + bd) \qquad (6)$$

$$AB^* - A^*B = j2(bc - ad) \qquad (8)$$

```
(defthm eq9-im-ab-star-is-crossp
  (equal (imz (cmul z (cconj w))) (crossp z w)))
```

Trigonometry as an `encapsulate`

Euler's formula and the dot product (eqs. 10–12); $\bullet$ is the planar dot product of eq. 7:

$$\cos\delta = \tfrac{1}{2}\left(e^{j\delta} + e^{-j\delta}\right) \tag{10}$$

$$\cos(\delta - \gamma) = \tfrac{1}{2}\left(e^{j\delta}e^{-j\gamma} + e^{-j\delta}e^{j\gamma}\right) \tag{11}$$

$$\cos(\delta - \gamma) = e^{j\delta} \bullet e^{j\gamma} \tag{12}$$

Genuinely trigonometric, so sine and cosine enter through an `encapsulate`:

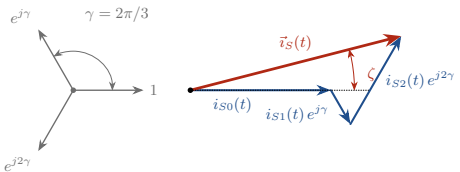
```
(encapsulate (((cosine *) => *)
              ((sine *) => *))
 (local (defun cosine (x) ... 1))
 (local (defun sine (x) ... 0))
 (defthm pythagorean-cos-sin ...)
 (defthm cosine-even ...)
 (defthm sine-odd ...)
 (defthm cosine-difference ...)
 (defthm sine-difference ...)
 (defthm acl2-numberp-cosine ...)
 (defthm acl2-numberp-sine ...))
```

- Plain ACL2 has no sine or cosine: results read *for any pair of functions with these properties*.
- Witness $\cos \equiv 1, \sin \equiv 0$: only consistency.
- In ACL2(r) all seven constraints are theorems about `acl2-cosine` and `acl2-sine`; functional instantiation gives eqs. 10, 12, the phase shift and the torque theorems for the real functions.

$\sqrt{3}$ as a free variable — and a vacuity caveat

- The roots (eqs. 18–19, $\gamma = 2\pi/3$ of eq. 1): $e_0 = 1$, $e_1 = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$, $e_2 = -\frac{1}{2} - j\frac{\sqrt{3}}{2}$, with $1 + e^{j\gamma} + e^{j2\gamma} = 0$, need $\sqrt{3}$.
- Plain ACL2 has no number with square 3: we use a variable `s3` with the hypothesis `(equal (* s3 s3) 3)`.
- A PC member pointed out that this hypothesis is unsatisfiable in plain ACL2, so those theorems are vacuously true there.
- What we changed:
  - `(rationalp s3)` → `(real/rationalp s3)`: rational in ACL2, real in ACL2(r);
  - the same books certify under both executables;
  - in ACL2(r), `(acl2-sqrt 3)` satisfies it, and the three theorems are instantiated there *without* any hypothesis on `s3`.

The space-vector angle (eqs. 13–28)




```
(defun ik (i0 i1 i2 s3) ...)
(defthm eq16-28-re-ik-balanced ...) ; (23)
(defthm eq16-28-imz-ik ...)         ; (21)
(defthm eq-ik-magnitude-squared ...); (27)
```

$$\sum_n i_{kn} = 0 \qquad (13)$$

$$\vec{i}_k = \sum_n i_{kn} e^{jn\gamma} = i_k e^{j\zeta_k} \qquad (16), (17)$$

$$\operatorname{Re} \vec{i}_k = \tfrac{3}{2}\, i_{k0} \qquad (23)$$

$$\operatorname{Im} \vec{i}_k = \tfrac{\sqrt{3}}{2}\,(i_{k1} - i_{k2}) \qquad (21)$$

$$\zeta_k = \operatorname{atan2}\big(\tfrac{\sqrt{3}}{2}(i_{k1} - i_{k2}),\ \tfrac{3}{2} i_{k0}\big) \qquad (26)$$

$$i_k = \tfrac{3}{2}\sqrt{i_{k0}^2 + \tfrac{1}{3}(i_{k1} - i_{k2})^2} \qquad (27)$$

Checked: (21), (23), (27). A $\frac{1}{3}$ vs. $\frac{1}{9}$ slip would hide in (27). We stop before (26): nothing more to check beyond the two components.

The three-phase collapse lemma (eqs. 33–35)

For any planar vector $\vec{A}$ constant in n (eq. 34a):

$$\sum_{n=0}^{2} \tfrac{2}{3}\left(\vec{A}\cdot e^{jn\gamma}\right) e^{jn\gamma} = \vec{A}$$

```
(defthm eq33-35-three-phase-collapse-lemma
  (implies (and (real/rationalp ra) (real/rationalp ia) (real/rationalp s3)
                (equal (* s3 s3) 3))
           (equal (cadd (cscale (* 2/3 (dotp (cplx ra ia) (e0))) (e0)) ...)
                  (cplx ra ia))))
```

With the phase shift $\vec{B}\cdot e^{j(n\gamma-\varphi)} = \left(\vec{B}e^{j\varphi}\right)\cdot e^{jn\gamma}$ (34g; no Pythagoras needed), the flux-linkage sum (34)

$$\vec{\psi}_k = \sum_n \tfrac{2}{3}\left\{ L_k\vec{i}_k\cdot e^{jn\gamma} + M\vec{i}_{k+1}\cdot e^{j(n\gamma-(-1)^k\varepsilon)}\right\} e^{jn\gamma} \;=\; L_k\vec{i}_k + M\vec{i}_{k+1}e^{j(-1)^k\varepsilon} \quad \text{(eq. 35)}$$

Torque: from Leonhard's integral to the cross product (eqs. 40–46)

$$T_M = \frac{M}{3\pi}\int_0^{2\pi} \frac{i_S i_R^* e^{-j\varepsilon} - i_S^* i_R e^{j\varepsilon}}{2j}\, d\beta \qquad (40)\ \text{Leonhard}$$

$$= \frac{2}{3} M \frac{i_S i_R^* e^{-j\varepsilon} - i_S^* i_R e^{j\varepsilon}}{2j} \qquad (41)\ \text{integrand constant in } \beta$$

$$W := i_S i_R^* e^{-j\varepsilon}, \quad W^* = i_S^* i_R e^{j\varepsilon}, \quad \frac{W - W^*}{2j} = \operatorname{Im} W \qquad (41a, 41b)\ \text{ACL2}$$

$$T_M = \frac{2}{3} M \operatorname{Im}\left[i_S \left(i_R e^{j\varepsilon} \right)^* \right] \qquad (42)\ \text{Leonhard}$$

$$\vec{T}_M = \frac{2}{3} M \left[\vec{i}_S \times \vec{i}_R\, e^{+j\varepsilon} \right] \qquad (46)\ \text{ACL2}$$

- Red: taken from Leonhard (2nd ed. (10.36), 3rd ed. (10.48)); not mechanized.
- Green: checked by ACL2, from W to the cross product, with identity (9) for $A = i_S$, $B = i_R e^{j\varepsilon}$.

The torque sign: what ACL2 checks

From $W(t) := i_S\, i_R^*\, e^{-j\varepsilon}$ as eq. 42 gives it:

```
(defun torque-W (i-s i-r eps) (cmul i-s (cconj (cmul i-r (eiz eps)))))

(defthm eq-im-w-is-crossp
  (equal (imz (torque-W i-s i-r eps)) (crossp i-s (cmul i-r (eiz eps)))))
```

`eiz eps` $= e^{+j\varepsilon}$: the algebra gives the $+\varepsilon$ form of eq. 46.

- `torque-exponent-sign-is-plus` restates this identity under a name that records the conclusion.
- Not vacuous: `sign-matters-nonvacuous` shows $+\varepsilon$ and $-\varepsilon$ differ at $\varepsilon = 2\pi/3$, $\vec{i_S} = 1 + j$, $\vec{i_R} = 1$.
- Scope: it checks eq. 42 $\rightarrow$ eq. 46. Eq. 42 itself is Leonhard's (2nd ed. (10.36), 3rd ed. (10.48)); not mechanized.

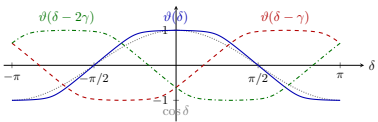
A more general collapse lemma, via ACL2(r)

- For *any* ω with $\omega^2 + \omega + 1 = 0$, not only the three roots.
- In plain ACL2 it is conditional: no rational complex number satisfies the cubic.
- In ACL2(r) we construct $\omega = e^{j2\pi/3}$ from `acl2-cosine`/`acl2-sine`:

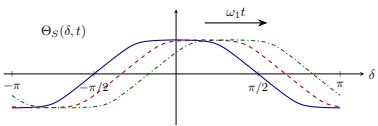
$$\cos\left(\tfrac{\pi}{3} + \tfrac{\pi}{3}\right) = \left(\tfrac{1}{2}\right)^2 - \left(\tfrac{\sqrt{3}}{2}\right)^2 = -\tfrac{1}{2}$$

  and `trinary-collapse-CONCRETE` holds with no hypothesis on ω.
- Two obstacles worth knowing:
  - `arithmetic-5` rewrites `(* x x)` to `(expt x 2)`: a bridge lemma, disabled right after use, or it loops;
  - a degenerate branch with real ω: closed by $x^2 + x + 1 > 0$ given as a `:use` hint.

The air-gap integral: open in the paper, closed since



winding functions $\vartheta(\delta - n\gamma)$ (28b)



traveling mmf wave $\Theta_S(\delta, t)$ (28a)

$$\psi_{kn} = \tfrac{N_k}{2}\, r\ell \int_0^{2\pi} B_k \cos(\delta_k - n\gamma)\, d\delta_k \qquad (29d)$$

$$\int_0^{2\pi} \cos(\delta - m\gamma)\cos(\delta - n\gamma)\, d\delta = \pi \cos[(m-n)\gamma] \qquad (29e)$$

$$\tfrac{1}{3}L_k = \tfrac{(1+\sigma_k)N_k^2 \ell r \pi \mu_0}{8h}, \qquad \tfrac{1}{3}M = \tfrac{N_k N_{k+1} \ell r \pi \mu_0}{8h} \qquad (30,\ 31)$$

Antiderivative of (29e), by hand:
$F(x) = \frac{1}{2}\cos[(n-m)\gamma]\,x + \frac{1}{4}\sin[2x - (m+n)\gamma]$.

- Plain ACL2 cannot state the integral; ACL2(r) can, through `ftc-2`. In the paper: algebra and raw derivative done; `acl2-sine/acl2-cosine` not.
- After the camera-ready, an AI agent registered them with the Reid–Gamboa differentiator (`sin-cos-minimal`), instantiated `ftc-2` and proved (29e), $\gamma = 2\pi/3$, for standard real m, n: `inductance-integral-value`, certified in ACL2(r).
- Not in the paper's books: branch `ftc2-inductance-integral`, tag `v1.1-ftc2`.

Can the paper alone reproduce the results?

The chairs plan an experiment: reproduce the results from the paper only. Before the final version we tried it.

- A fresh agent, given only the PDF of the accepted paper, offline ACL2 in Docker, no access to the repository.
- It certified everything the paper claims, in ACL2 and ACL2(r).
- What it had to guess, now in the final version:
  - the definitions of e_0, e_1, e_2 and `ik`, and the exact statements of eqs. 10 and 12;
  - the flux-linkage sum behind eq. 35;
  - names that did not match the books (`im`/`imz`, two theorem names), and two type facts missing from the printed `encapsulate`.
- It also noted that the torque theorem restates another one — now said plainly.

How the books were developed

- With Claude Code, an AI coding agent, driving ACL2 and ACL2(r) through an MCP server (`admit`, `prove`, `certify_book`).
- The loop is the usual one: state a `defthm`, let ACL2 accept or reject it, adjust, retry.
- The author reviewed every theorem statement.
- ACL2 is the arbiter: every `defthm` is certified by `cert.pl` from a clean state, no matter who proposed the script.

Summary

- Every algebraic step from the complex form to the vectorial model is checked, up to the $+j\varepsilon$ sign of the torque equation.
- Trigonometry as axioms in ACL2, shown to hold in ACL2(r); $\sqrt{3}$ handled honestly in both.
- A general collapse lemma, unconditional in ACL2(r).
- The one calculus step, the air-gap integral, was open in the paper; it was closed afterwards, outside the paper's books (tag `v1.1-ftc2`).

Books: `[books]/workshops/2026/cayuela-ac-machine/`
Development version: `github.com/pcayuela/ACL2-ACmotormodel-check`
Appendix (PDF):
`github.com/pcayuela/ACL2-ACmotormodel-check/blob/7d8def9/docs/appendix-full-derivation.pdf`

Thank you — questions?

Backup: the generic and the concrete ω in ACL2(r)

```
(defun omega-real ()
  (cplx (acl2-cosine (gamma-val)) (acl2-sine (gamma-val))))

(defthm omega-real-satisfies-cubic
  (equal (c+ (cplx 1 0) (c+ (omega-real) (c* (omega-real) (omega-real))))
         (cplx 0 0)))

(defthm trinary-collapse-CONCRETE
  (implies (realpairp A)
           (equal (collapse-sum A (omega-real)) A)))
```

Backup: how the books are certified

Top book	Contents
`top`	00–07, the main chain (plain ACL2 or ACL2(r))
`top-acl2r`	the generic ω in ACL2(r); the inductance-integral progress
`top-acl2r-nonvacuity`	09: constraints, $\sqrt{3}$ instances, functional inst.

All functions have verified guards, except those that call `acl2-pi`, `acl2-sine` or `acl2-cosine`.