Optimization of Boost Converter Controller Design

Kevin Fronczak

The Problem

- Boost Converters can be difficult control
 - Need stability (Phase Margin)
 - Need low steady-state error (Open Loop Gain)
 - Need fast response (Bandwidth)
- Two operating modes:
 - CCM (Current always flowing in inductor)
 - DCM (Current in inductor goes to zero within a switching cycle)

L

Vramp(M)

Vref

<R⊤

≤R_B

 \mathbf{Vs}



The Problem Controller Topologies

• Lag Controller

$$\frac{V_c}{V_o} = K \left[\frac{sR_ZC_Z + 1}{sR_OC_Z + 1} \right]$$
$$K = g_m R_O \frac{R_B}{R_T + R_B}$$

Lag Plus Pole Controller

$$\frac{V_c}{V_o} = K \frac{sR_ZC_Z + 1}{s^2R_OR_ZC_CC_Z + sR_OC_Z + 1}$$

$$K = g_m R_O \frac{R_B}{R_T + R_B}$$

• Lag-Lead Controller (a.k.a. PID)

$$\frac{V_c}{V_o} = \frac{g_m R_B}{s(R_B + R_T)(C_Z + C_C)} \frac{(sR_T C_1 + 1)(sR_Z C_Z + 1)}{(s\frac{R_Z C_Z C_C}{C_Z + C_C} + 1)(s(R_T || R_B)C_1 + 1)}$$



Proposed Solution Genetic Algorithms and PSO

- Controller Design \rightarrow Dependent on Converter Design
 - Operating mode
 - Desired characteristics
- GAs
 - Good exploration of solution space
 - Large amount of previous work on GAs for Circuit Design Optimization
 - Highly dependent on fitness function, mutation probability, crossover probability
- PSOs
 - Good exploration of solution space
 - Tendency to get "stuck" on local optima

Fitness Function

Implementation Comparisons

- Parameters with large ranges
 - Need to normalize
- Two functions:
 - Linear
 - Parabolic





Fitness Function Selected Implementation

- Penalty function added
 - Aids convergence
- Steps:
 - Normalize variable → Calculate penalty → Place in Parabola → Apply weighted constant



Genetic Algorithm Queen Bee Architecture

- Queen Bee Implementation
 - Variant on elitism



Genetic Algorithm Mutation Rate

- Attempt: Mutation rate that varies with Queen age
 - As Queen's age increased, mutation rate probability increased

- Variable rate:
 - Convergence in 11 iterations
- Static rate:
 - Convergence in 25 iterations



Note: 30 drones simulated for DCM Boost

Particle Swarm Optimization

Swarm constricted to smaller solution space over time

$$V_p = \chi (V_p + C_1 [U(0, 1) \cdot (\beta_p - X_p)] + C_2 [U(0, 1) \cdot (G - X_p)])$$

$$\phi = C_1 + C_2$$

$$\chi = \frac{2}{2 - \phi - \sqrt{\phi^2 - 4\phi}}$$

Particle Swarm Optimization Chaotic Inertial Weight

- Inertial Weight with Randomness
 - Limit particle's speed over time
- Chaotic Descending (CDIW)

$$z = 4 \cdot z \cdot (1 - z)$$
$$w = \frac{(w_1 - w_2)(i_{max} - i)}{i_{max}} + z \cdot w_2$$

Chaotic Random (CRIW)

z

w

$$= 4 \cdot z \cdot (1 - z)$$

= $\frac{1}{2} [U(0, 1) + z]$
$$\geq 0.5$$

0
0
100
100
200
300
Iteration
CRIW
$$\geq 0.5$$



TTTA7

10

500

Algorithm Comparisons Convergence

- GA performed better
- PSO consistently converged on local optima
- Constriction better than Chaotic IW

	Fitness Convergence Comparison							
FitnessValue	5 0 -5 -10 -15 -20	 - ● - ● - ● 				GA Constrict IW		
		20	40 Ite	6 eration	0 8	30 1	.00	

Algorithm	Error From Expected
QBGA (Static)	0.268%
QBGA (Variable)	0.179%
PSO-CDIW	133.286%
PSO-CRIW	133.036%
PSO-Constrict	73.750%

Algorithm Comparisons DCM Solutions

$$T_{QBGA}(s) = \frac{0.0034s + 159.8}{0.0026s + 1}$$
$$T_{PSO}(s) = \frac{147s^2 + 288.9s + 3.016}{0.196s^3 + 0.0066s^2 + 0.000021s + 0.00002}$$



Algorithm Comparisons CCM Solutions

$$T_{QBGA}(s) = \frac{348.6s + 348.6}{0.00017s^3 + s^2 + 3.5 \times 10^{-7}s + 1}$$
$$T_{PSO}(s) = \frac{4.2 \times 10^4 s + 2.02 \times 10^4}{0.06s^3 + 48.53s^2 + 0.27s}$$



Both valid solutions

- QBGA better large-signal response

Summary

- Variable mutation rate for QBGA yields better convergence
- Parabolic parameterization aids convergence
- GA better suited for Boost Converter controller design
 - PSO Constrict better than Chaotic Inertial Weight
- Problems encountered:
 - Selection of weighted constants in fitness function = tedious
 - Experimentation and iteration
 - PM \rightarrow 3; Gain \rightarrow 2; GM \rightarrow 0.6
 - CCM solutions harder to find
 - Expected since it's harder to control → requires more iterations than DCM



Thank You

• Comments/Questions?