Output Voltage Filtering in Pulse Width Modulation Based D/A Converters

Predrag Pejović

introduction

- ▶ PWM D/A converters ...
- ▶ microcontrollers and (subset) Arduino ...
- ▶ sometimes filtering not needed (light dimming) ...
- ▶ sometimes needed (voltage reference)
- ▶ solution readily available?
- ▶ **not!**
- ▶ “complicated math” ... ?!
- ▶ I needed a recipe-like solution ...
- ▶ n and f_S determine the filter parameters ...
- ▶ requirement: $V_{ripple\ p-p} < 1\ V_{LSB}$
- ▶ $\Rightarrow$ filter structure and parameters, transient time ...
- ▶ math should not be a problem (linear, scalable over f_S) ...
- ▶ methods: small ripple approximation (SRA, LRA, PRA) and sinusoidal approximation ...

pulse-width modulation

$$V_{PWM} = D V_{ref}$$

how to filter the mean value and how to get rid of the ripple?

worst case analysis, sinusoidal approximation

- ▶ $V_{PWM,1} = \frac{2}{\pi} V_{ref} \sin(\pi D)$
- ▶ maximum $V_m = \frac{2}{\pi} V_{ref}$ at $D = \frac{1}{2}$
- ▶ the second harmonic $V_{PWM,2} = \frac{1}{\pi} V_{ref} |\sin(2\pi D)|$
- ▶ at $D = \frac{1}{2}$ vanishes
- ▶ otherwise, $\frac{1}{2}$ in amplitude ...
- ▶ sinusoidal approximation: let's consider the first harmonic, only!
- ▶ filter independent!

RC filter of the first order

linear ripple approximation

linear ripple approximation (LRA)

- ▶ $\Delta v_{OUT} = \frac{D(1-D)}{2f_S RC} V_{ref}$

- ▶ maximum for $D = \frac{1}{2}$

- ▶ $\Delta v_{OUT,max} = \frac{V_{ref}}{8f_S RC}$

- ▶ $\Delta v_{OUT,max} = \frac{V_{ref}}{2^{n+1}}$

- ▶ $\frac{V_{ref}}{8f_S RC} = \frac{V_{ref}}{2^{n+1}}$

- ▶ $\omega_P = \frac{1}{RC} = \frac{f_S}{2^{n-2}}$

- ▶ this **is** a recipe!

- ▶ so $\omega_P = \frac{f_S}{2^{n-2}}$

sinusoidal approximation

- ▶ $H_1(s) = \frac{1}{1 + \frac{s}{\omega_P}}$
- ▶ $\omega_P = \frac{1}{RC}$
- ▶ $\Delta v_{OUT} = \frac{\frac{2}{\pi} V_{ref}}{\sqrt{1 + \left(\frac{2\pi f_S}{\omega_P}\right)^2}}$
- ▶ $\omega_P = \frac{1}{RC} = \frac{2\pi^2 f_S}{\sqrt{2^{2(n+2)} - \pi^2}}$
- ▶ $\omega_P \approx \frac{\pi^2}{8} \frac{f_S}{2^{n-2}}$
- ▶ different, 23.37% larger then before; why?
- ▶ better, but worse! keep the LRA result!

transient response of the first order filter

- ▶ in the worst case there is no ripple!
- ▶ $v_{OUT\,err}(t) = v_{OUT}(t) - v_{OUT}(\infty)$
- ▶ $|v_{OUT\,err}(t)| \leq \frac{1}{2^{n+1}} V_{ref}$
- ▶ rules of thumb $t_T = 3\tau_P$ or $t_T = 5\tau_P$ for the error within 5% and 1%, are not applicable, $\tau_P = 1/\omega_P$
- ▶ $|v_{OUT\,err}(t)| = V_{ref} e^{-\omega_P t}$
- ▶ $t_T = \frac{n+1}{\omega_P} \ln(2)$
- ▶ for other transients ...
- ▶ $t_T = \frac{1}{\omega_P} (\ln(p) + (n+1) \ln(2)), 0 < p \leq 1$
- ▶ quite computable ...

second order filters, general

- ▶ two parameter transfer function, ω_P and Q_P

- ▶
$$H_2(s) = \frac{1}{1 + \frac{s}{Q_P \omega_P} + \frac{s^2}{\omega_P^2}}$$

- ▶ a special case, $Q_P = \frac{1}{2}$

- ▶
$$H_2(s) = \frac{1}{\left(1 + \frac{s}{\omega_P}\right)^2}$$

- ▶ $Q_P = \frac{1}{2}$ results in the fastest transient with no overshoot, the lowest setting time, optimal damping

RC filter of the second order

RC filter of the second order

- ▶ $H_2(s) = \frac{1}{1 + \frac{1+2a}{a} RC s + (RC)^2 s^2}$
- ▶ $\omega_P = \frac{1}{RC}$
- ▶ $Q_P(a) = \frac{a}{1 + 2a}$
- ▶ $\lim_{a \rightarrow \infty} Q_P(a) = \frac{1}{2}$
- ▶ a special case of $a = 1$, $Q_P(1) = \frac{1}{3}$
- ▶ another special case, $a = 10$, $Q_P(10) = 0.47619$,
in this application $10 \approx \infty$

RLC filter of the second order

RLC filter of the second order

- ▶ $H_2(s) = \frac{1}{1 + RCs + LCs^2}$
- ▶ $\omega_P = \frac{1}{\sqrt{LC}}$
- ▶ $Q_P = \frac{1}{R} \sqrt{\frac{L}{C}}$
- ▶ $R = 2 \sqrt{\frac{L}{C}}$ to provide optimal damping

ω_P for $Q_P = \frac{1}{2}$, LRA, actually LRA + PRA, based on SRA

$$\blacktriangleright \Delta v_{OUT, max} = \frac{V_{ref}}{64 f_S^2 R^2 C^2}$$

$$\blacktriangleright \frac{V_{ref}}{64 f_S^2 R^2 C^2} = \frac{V_{ref}}{2^{n+1}}$$

$$\blacktriangleright f_S R C = 2^{\frac{n-5}{2}}$$

$$\blacktriangleright \omega_P = \frac{f_S}{2^{\frac{n-5}{2}}}$$

$$\blacktriangleright 2^{\frac{n+1}{2}} \text{ times higher than before!}$$

$$\blacktriangleright \text{try } Q_P = \frac{1}{3} \text{ and you'll get the same result!}$$

$$\blacktriangleright \text{so } \boxed{\omega_P = \frac{f_S}{2^{\frac{n-5}{2}}}}$$

ω_P for $Q_P = \frac{1}{2}$, SA

- ▶ $\Delta v_{OUT} = \frac{\frac{2}{\pi} V_{ref}}{1 + \left(\frac{2\pi f_S}{\omega_P}\right)^2}$
- ▶ $\omega_P = f_S \sqrt{\frac{4\pi^3}{2^{n+2} - \pi}}$
- ▶ $\lim_{n \rightarrow \infty} \frac{\pi \sqrt{\pi}}{4\sqrt{2}} \approx 0.9844$
- ▶ almost the same, as expected!

ω_P for $Q_P = \frac{1}{3}$, SA

► doable!

► $|H_2(j 2 \pi f_S)| \frac{2}{\pi} V_{ref} = \frac{V_{ref}}{2^{n+1}}$

► wxMaxima ...

►
$$\omega_P = f_S \sqrt{\frac{8 \pi^3}{\sqrt{2^{2n+6} + 45 \pi^2} - 7 \pi}}$$

► more of the same: as $n \rightarrow \infty$ relation to LRA, PRA, SRA
 $\frac{\pi}{4} \sqrt{\frac{\pi}{2}} \approx 0.9844$, the same as for $Q_P = \frac{1}{2}$

► conclusion: case $Q_P = \frac{1}{2}$ is sufficient to determine ω_P !

► so we already have the boxed equation for the 2nd order

transient duration, second order filters

- ▶ awful equations
- ▶ different for $Q_P = \frac{1}{2}$ and for $Q_P = \frac{1}{3}$
- ▶ you have the equations in the paper ...
- ▶ to determine t_T you end up with transcendental equations
- ▶ something like $(1 + \omega_P t_T) e^{-\omega_P t_T} = \frac{1}{2^{n+1}}$
- ▶ or like $\frac{3 + \sqrt{5}}{2\sqrt{5}} e^{-\frac{3-\sqrt{5}}{2} \omega_P t} - \frac{3 - \sqrt{5}}{2\sqrt{5}} e^{-\frac{3+\sqrt{5}}{2} \omega_P t} = \frac{1}{2^{n+1}}$
- ▶ to compute $\omega_P t_T$ the only solution is a numerical solution
- ▶ so I did it!

transient duration, second order filters, numerical solutions

n	$\omega_P t_T, Q_P = \frac{1}{2}$	$\omega_P t_T, Q_P = \frac{1}{3}$
4	5.3075	9.4863
5	6.1221	11.3010
6	6.9216	13.1157
7	7.7096	14.9303
8	8.4884	16.7450
9	9.2597	18.5597
10	10.0248	20.3744
11	10.7846	22.1891
12	11.5398	24.0038

another rule of thumb:

$Q_P = \frac{1}{2}$ is about twice faster than $Q_P = \frac{1}{3}$

rule of thumb: computing ω_P , scaling coefficients

$\omega_P = \frac{f_S}{s_k}$, k is the filter order, from boxed equations

n	s_1	s_2
4	4	0.71
5	8	1.00
6	16	1.41
7	32	2.00
8	64	2.83
9	128	4.00
10	256	5.66
11	512	8.00
12	1024	11.31

either $\omega_P = \frac{1}{RC}$, either $\omega_P = \frac{1}{\sqrt{LC}}$

rule of thumb: setting time, t_T

n	t_{T1}/T_S	$t_{T2}/T_S, Q_P = \frac{1}{2}$	$t_{T2}/T_S, Q_P = \frac{1}{3}$
4	13.86	3.75	6.71
5	33.27	6.12	11.30
6	77.63	9.79	18.55
7	177.45	15.42	29.86
8	399.25	24.01	47.36
9	887.23	37.04	74.24
10	1951.90	56.71	115.25
11	4258.70	86.28	177.51
12	9227.18	130.56	271.57

$$\text{either } Q_P = \frac{a}{1+2a}, \text{ either } Q_P = \frac{1}{R} \sqrt{\frac{L}{C}}$$

experimental results

- ▶ Arduino Uno
- ▶ $f_S = 490.206 \text{ Hz}$, $T_S = 2.04 \text{ ms}$
- ▶ LSB quantum of 19.53 mV
- ▶ $n = 8$, $\tau_{P1} = \frac{1}{\omega_{P1}} = 130.56 \text{ ms}$, $\tau_{P2} = \frac{1}{\omega_{P2}} = 5.7731 \text{ ms}$
- ▶ 1st order: $R = 1 \text{ k}\Omega$ and $C = 147 \mu\text{F}$
- ▶ 2nd order, 1: $R = 1 \text{ k}\Omega$, $C = 6.8 \mu\text{F}$, $a = 1$, $Q_P = \frac{1}{3}$
- ▶ 2nd order, 2: $R = 1 \text{ k}\Omega$, $C = 6.8 \mu\text{F}$, $a R = 10 \text{ k}\Omega$,
 $C/a = 680 \text{ nF}$, $a = 10$, $Q_P \approx 0.47619 \approx \frac{1}{2}$
- ▶ $t_{T1} = 814.47 \text{ ms}$, $t_{T2} = 96.614 \text{ ms}$, $t_{T3} = 48.98 \text{ ms}$
- ▶ did I get what I predicted?

output voltage ripple, filter 1

voltage scale 5 mV/div; time scale 500 μ s/div

output voltage ripple, filter 2

voltage scale 5 mV/div; time scale 500 μ s/div

transient response, filter 1

yellow: filtered output voltage; cyan: D/A converter command;
time scale: 500 ms/div

transient responses of filters 2 and 3

2—yellow, 3—cyan; time scale: 25 ms/div

conclusions

- ▶ PWM D/A converters, low end
- ▶ filtering ...
- ▶ to keep the ripple low enough ...
- ▶ to provide the transient as fast as possible ...
- ▶ first order and second order filters ...
- ▶ three structures ...
- ▶ methods: LRA, PRA, SRA, SA, ...
- ▶ for the second order predictions are the same ...
- ▶ some difficulties to compute the transient time ...
- ▶ **rules of thumb** for ω_P and for t_T ...
- ▶ experimentally verified
- ▶ second order filters are much faster!