

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پردازش بی درنگ سیگنال‌های دیجیتال

تمرین شماره ۶

نام و نام خانوادگی : سهند خوشدل

شماره دانشجویی : ۸۱۰۱۹۶۶۰۷

پاییز ۹۹

بخش ۲-۴-۷:

الف) جدول نتایج تخمین مقدار **cosine** به کمک توابع داده شده :

The cosine estimated values using the 3 functions are:

theta	fcos1(theta)	fcos2(theta)	icos(theta)
0.000000	1.000000	1.000000	0.999969
0.100000	0.995004	0.995004	0.995026
-0.800000	0.696703	0.696703	0.696716
19.373156	-67746.656250	-67746.656250	0.809906
-0.523592	0.866029	0.866029	0.865967
1.570796	-0.000894	-0.000894	-0.000916
3.141592	-1.211352	-1.211353	0.653595

$\theta(rad)$	$\cos(\theta)$	$fcos1(\theta)$	$fcos2(\theta)$	$Icos(\theta)$
0	1.0000	1.000000	1.000000	0.999969
0.1	0.9950	0.995004	0.995004	0.995026
- 0.8	0.6967	0.696703	0.696703	0.69716
19.373155	0.8660	- 67746.656250	- 67746.656250	0.809906
0.523592-	0.8660	0.866029	0.866029	0.865967
1.5707963	0.0000	- 0.000894	- 0.000894	- 0.000916
3.141592	-1.0000	- 1.211352	- 1.211353	0.653595

مشاهده میکنیم که تفاوت چندانی بین خروجی های $fcos1, fcos2$ به ازای زوایای بالا ایجاد نشد و همچنین $icos$ نسبت به زوایایی که زاویه اصلی آن ها (باقی مانده تقسیم زاویه به 2π مقدار کوچکی است) خوب عمل میکند اما دو تابع اول نتایج نادرستی می دهند.

ب) پاسخ سوالات دستور کار :

سوال: چرا آرگومان کسینوس برای fcos1 نباید عددی بزرگ باشد؟

پاسخ:

چون این تابع بر اساس پیاده سازی بسط تیلور تابع کسینوس حول نقطه $x = 0$ (بسط مکلاورن تابع \cos)، ضرایب را بدست می آورد و محاسبات مربوطه را انجام می دهد. ضرایب بسط تیلور حول اعداد بزرگتر اختلاف نسبتا زیادی از ضرایب بسط مک لورن دارند، که این عامل موجب افزایش قابل توجه خطا می شود.

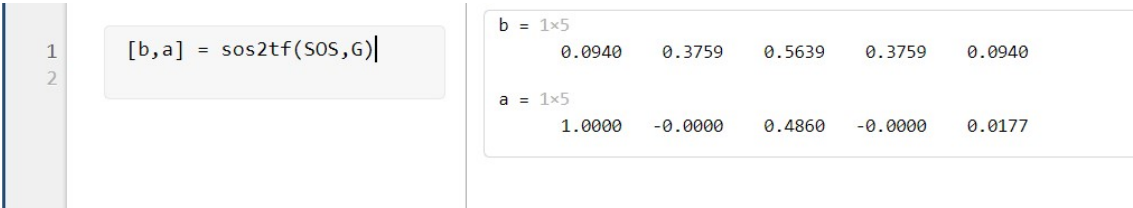
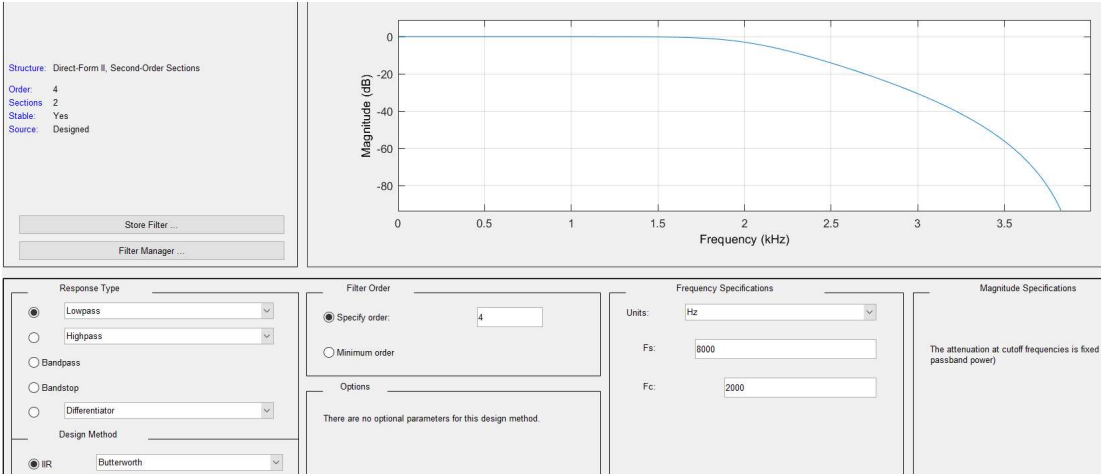
سوال: چرا در بدنه تابع acos متغیر \cos پس از هر مرحله ضرب شدن جملات در ضرایب ۱۳

واحد به سمت راست شیفت می شود؟

پاسخ:

برای افزایش دقت محاسبات هر جای کد هنگامی که دو عدد از نوع short در هم ضرب شده اند ابتدا خروجی در یک متغیر از نوع long ذخیره شده است. (چرا که ورودی تابع icos که همان x است از نوع Q2.14 (دارای ۱۴ رقم اعشار) است و ضرایب بسط تیلور نیز دارای ۱۵ رقم اعشار هستند و حاصلضرب این دو می تواند تا ۳ رقم صحیح و ۲۹ رقم اعشار داشته باشد که نیاز به ۳۲ بیت برای ذخیره سازی دارد) از طرفی می دانیم که خروجی تابع \cos باید عددی بین صفر و یک باشد که در صورت ذخیره سازی اعداد float به صورت fixed point ۱۶ بیت ابتدایی به قسمت صحیح و ۱۶ بیت انتهایی به قسمت اعشاری تعلق دارد بنابراین برای درست ذخیره شدن مقدار \cos قبل از جمع کردن آن با جمله بعدی نیاز به ۱۳ واحد شیفت به سمت راست داریم .

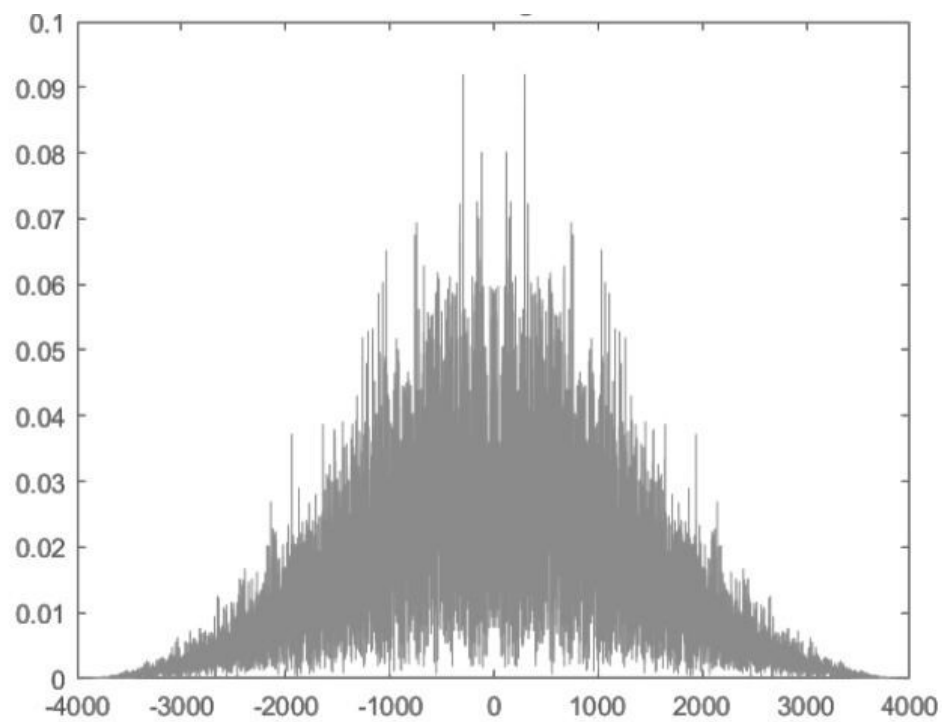
الف) تولید فیلتر IIR در متلب و بدست آوردن ضرایب:



فیلتر IIR مطابق دستور کار از مرتبه ۶ با فرکانس قطع 2 kHz و فرکانس نمونه برداری 8 kHz و از نوع Butterworth طراحی شد و ضرایب به صورت SOS, G استخراج شدند. برای پیاده سازی راحت تر فیلتر در c ضرایب به کمک دستور sos2tf به شرایب استاندارد a,b تبدیل شدند.

برای پیاده سازی فیلتر به صورت fixed point، باید ورودی، ضرایب و خروجی تابع IIR_fixed به صورت fixed_point باشد. با توجه به این نکته که ورودی عددی بین -۱ و ۱ است برای نمایش آن ۱ رقم صحیح کافی است و ورودی را به صورت Q1.15 دریافت می کنیم. ضرایب هم به فرمت Q2.14 خوانده شده اند، خروجی پس از خوانده شدن از تابع IIR_fixed، ۱۵ واحد شیفت پیدا کرده و به صورت

Float در آمده و ذخیره می شود. سپس در نرم افزار متلب به رسم طیف سیگنال خروجی فیلتر شده می پردازیم.



مشاهده می کنیم که فیلتر نسبت به فیلتر طراحی شده در متلب ضعیف تر عمل میکند (به خاطر fixed point بودن) ، اما کاهش دامنه طیف در حدود ۲ کیلو هرتز نسبتاً مشخص است.