

Le FS30-DCF – Un standard de fréquence pour les bandes “courant continu” (1/2)

De FS30-DCF – Een frequentiestandaard voor de gelijkstroombanden (1/2)

Par/door PA0WV – Traduit par ON5WF

Introduction

Depuis longtemps, je caressais le rêve de disposer de mon propre standard de fréquence, de façon à pouvoir étalonner un fréquencemètre, connaître avec précision les limites des bandes HF, mesurer avec précision le coefficient de température d'un quartz et d'une manière générale, rendre plus performant l'outillage disponible. Pour les amateurs qui veulent récupérer des signaux dans un bruit important, un tel standard est un must.

Chez Baco Army Goods à Ijmuiden, j'ai acheté à Klove, pour une paire d'euros, quelques TCXO's pour 10 MHz. Cela m'a donné l'impulsion pour, enfin, en fabriquer un, en utilisant autant que possible les fonds de tiroir et stocks de composants. Finalement, j'en ai construit deux, le premier, décrit dans cet article, est basé sur DCF77, et le second, le FS30-GPS, est un tout autre projet basé sur le GPS.

Un quartz ordinaire pouvant être décalé d'une dizaine de hertz avec une varicap pourra être utilisé, surtout s'il s'agit d'une coupe AT ou SC et que le quartz est stabilisé en température, au moyen d'un four, sur un extremum ou un point d'inflexion de la courbe fréquence-température. En tout cas, cela incite à se lancer dans d'excitantes expériences.

Conception

Un TCXO est un oscillateur à quartz compensé en température. La température peut donc varier. La fréquence de résonance est rendue moins dépendante de la température ambiante grâce à des CTN et une varicap incorporée dans le boîtier. A l'une des broches du boîtier, on peut appliquer une tension continue qui permet de faire varier dans une certaine mesure la fréquence de résonance du quartz. On peut aussi utiliser cette broche pour asservir la fréquence de résonance du quartz à celle d'un émetteur étalon.

La gigue et les perturbations du même genre occasionnées par les différences entre les chemins suivis par l'onde lors de sa propagation, sont d'autant plus faibles que la longueur d'onde est grande. C'est pour cette raison que mon choix s'est fixé initialement sur DCF77 (77,5 kHz), mais aussi parce que je disposais encore dans mes fonds de tiroirs, d'un récepteur à amplification directe pour DCF77. J'avais réalisé ce récepteur suite à un article paru dans Radio Electronica n° 23 de 1975 pp 797 à 799, sous la plume de PA0JVK. Tout récepteur à amplification directe qui reçoit la fréquence de 77,5 kHz via une antenne ferrite résonnant sur 77,5 kHz, avec un quartz de 77,5 kHz comme élément sélectif et un AVC approprié, peut évidemment convenir.

L'objectif est que, si cet émetteur étalon tombe en panne suite à des perturbations locales ou pour cause d'orage, le standard de fréquence continue à fonctionner avec sa dernière tension de réglage. De même, dans le cas d'une coupure de la tension du réseau, le système doit continuer à fonctionner. En outre, l'objectif est de pouvoir connecter un phasemètre qui peut enregistrer la gigue, aussi bien à l'œil (avec le Zerobeater publié précédemment dans CQPA), qu'avec un PC qui stocke les mesures dans un fichier.



Inleiding

Het was hier een lang gekoesterde wens om een eigen frequentiestandaard ter beschikking te hebben, waarmee je je teller kunt ijken en je de bandgrenzen precies weet op de hf-banden, de temperatuurcoëfficiënt van kristallen kunt meten en dergelijke, zodat je met je beschikbare gereedschap beter gereedschap kunt maken. Voor de hams die signalen diep uit de ruis willen opdiepen is een standaard vrijwel een noodzaak.

Bij Baco Army Goods in Ijmuiden heb ik enkele TCXO's voor 10 MHz van Klove voor een paar euro gekocht. Dat gaf

het duwtje in de rug om er eindelijk eens een te gaan maken, zoveel mogelijk met spullen uit de junk box en de grijpvoorraad, waar heb je die immers anders voor. Uiteindelijk zijn er twee typen ontwikkeld, de eerste op DCF77, die hier nu beschreven wordt en de tweede, een geheel ander ontwerp, de FS30-GPS, gebaseerd op GPS.

Met een gewoon kristal dat met een varicap een tiental hertz verstemd kan worden zal ook wel mee te werken zijn, vooral als je er een AT-cut of SC-cut hebt en er een oventje voor maakt om de temperatuur te stabiliseren op een extremum of buigpunt van de temperatuur-frequentiecurve. In ieder geval daagt een en ander uit tot experimenteren en dat is waar het radioamateurhart, zelfs op de oude dag, sneller van gaat kloppen.

Ontwerp

Een TCXO is een temperatuurgecompenseerde kristaloscillator. De temperatuur mag dus verlopen. Door ingebouwde NTC's en een varicap wordt de resonantiefrequentie een stuk minder van de omgevingstemperatuur afhankelijk, stabiel gehouden. Op een inputpen kun je een gelijkspanning aanbieden die de resonantiefrequentie bepaalt en die je met die spanning wat kunt vertrekken. Je kunt die pen ook gebruiken om de frequentie automatisch te corrigeren als die afwijkt van de frequentie van een ijkzender.

Hoe langer de golflengte, hoe minder last ijkzenders hebben van jitter veroorzaakt door padlengteverschillen en dergelijke verstoringen op de propagatieweg. Daarom is aanvankelijk gekozen voor DCF77 op 77,5 kHz, mede omdat ik daar nog een rechthoekig ontvanger voor had liggen in de junkbox, lang geleden gebouwd naar een ontwerp uit Radio Electronica in het nr 23 van 1975 blz. 797 t/m 799, van de hand van PA0JVK. Elk rechthoekig ontvanger dat 77,5 kHz ontvangt, via een op 77,5 kHz resonerende ferrietstaafantenne met een 77,5 kHz kristal als selectief element en een passende AVC is uiteraard bruikbaar.

De bedoeling is, dat als die ijkzender uitvalt, door lokale storing of omdat hij uitgeschakeld wordt vanwege plaatselijk onweer, de standaard gewoon door blijft lopen met zijn laatst ingestelde regelspanning. Ook als tussendoor de netspanning wegvalt moet de zaak gewoon doorfietsen. Voorts is het de bedoeling dat een uitprintbare fasemeter, die de jitter kan registreren zowel met het oog (met de eerder in CQPA gepubliceerde Zerobeater) als met een daarop aangesloten PC die een bestand maakt van de meetwaarden, kan worden aangesloten.

Commençons par réfléchir.

La fréquence de 10 MHz du quartz peut être décomposée selon le produit $2^7 \times 5^7$ et la fréquence de DCF77 vaut 77,5 kHz, ou $2^2 \times 5^4 \times 31$. Il n'y a donc pas de diviseur entier permettant d'obtenir, à partir de 10 MHz, une fréquence 77,5 kHz à comparer à celle de l'émetteur étalon.

On peut donc se demander s'il existe deux entiers m et n tels que $10^7 / m \pm 10^7 / n = 77500$. Dans ce cas, on peut obtenir, par division de la fréquence du quartz, deux fréquences qui donneront alors 77,5 kHz par mélange.

Il existe en fait plusieurs couples de tels entiers:

- $m = 375$ et $n = 96$
- $m = 128$ et $n = 128 \times 125 = 16000$ et aussi
- $m = 125$ et $n = 32 \times 125 = 4000$

On a donc le choix.

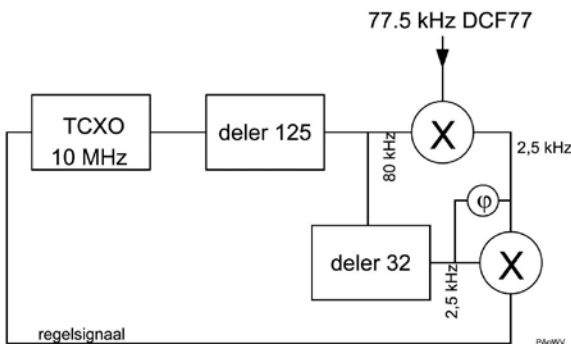


Fig. 1

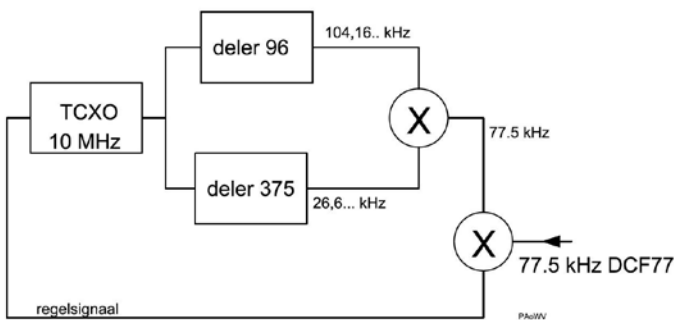


Fig. 3

Tout cela est illustré dans les **figures 1 à 5**. Dans le premier cas $m = 375$ et $n = 96$ (**figure 3**), on a besoin pour les diviseurs, de 5 CI discrets de la série 74HCT, et de 4 pièces pour les deux cas suivants. De plus, dans le deuxième et le troisième cas, on a affaire à des CI à 14 broches au lieu de 16, ce qui libère de la place sur le circuit imprimé, lequel selon la loi pessimiste de Murphy, se révélera être trop petit lors de la réalisation.

Si l'on divise par 125 la fréquence de l'oscillateur à 10 MHz (voir le schéma bloc de la **figure 1**), on obtient 80 kHz. En mélangeant ces 80 kHz avec la fréquence de référence de 77,5 kHz, on obtient par différence une fréquence de 2,5 kHz. La fréquence de 80 kHz est également divisée par 32, ce qui donne 2,5 kHz. En mélangeant cette dernière fréquence avec celle issue du premier mélangeur, on obtient 0 Hz, soit une composante continue (signal d'erreur). Cette composante continue dont la grandeur dépend du déphasage entre le TCXO et la fréquence de référence de l'émetteur étalon, est utilisée pour asservir en fréquence et phase, le TCXO à DCF77.

We gaan beginnen met nadenken

De kristalfrequentie van 10 MHz is te ontbinden in priemfactoren $2^7 \times 5^7$ en de DCF77-frequentie van 77,5 kHz is $2^2 \times 5^4 \times 31$. Er is dus geen geheel deeltal om 10 MHz te delen tot 77,5 kHz dat dan vergeleken kan worden met de frequentie van de ijkzender.

Vervolgens kun je je afvragen of er dan 2 gehele getallen m en n te vinden zijn zodanig dat $10^7 / m \pm 10^7 / n = 77500$. Dat wil dus zeggen dat je twee frequenties door deling van de kristalfrequentie maakt, die mengt en dan op 77,5 kHz uitkomt.

Die zijn er inderdaad:

- $m=375$ en $n=96$
- $m=128$ en $n=128 \times 125=16000$ en ook
- $m=125$ en $n=32 \times 125= 4000$

We kunnen dus uitkiezen.

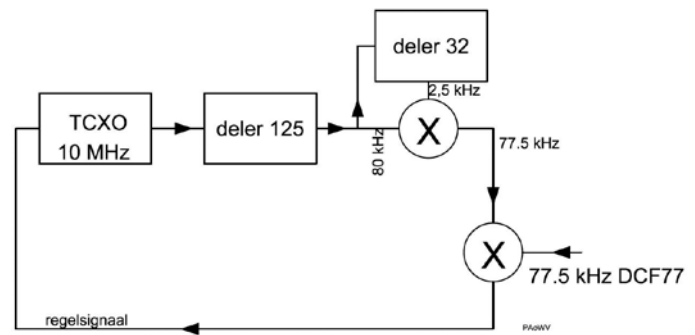


Fig. 2

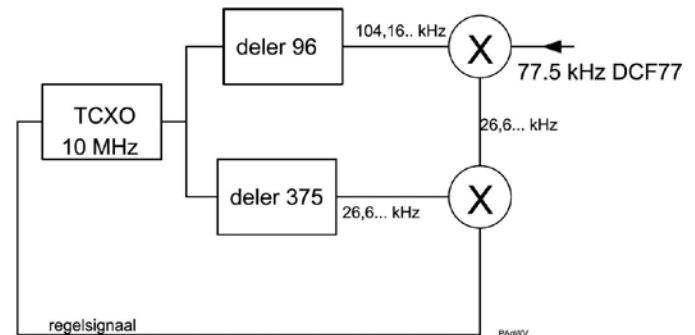


Fig. 4

Een en ander is toegelicht in tekeningen in de **figuren 1 t/m 5**. In het eerste geval $m=375$ en $n=96$ (**figuur 3**) heb je voor de delers 5 discrete IC's nodig uit de 74HCT serie, en in beide volgende gevallen 4 stuks, om die deeltallen te realiseren. Bovendien zijn dat in het tweede en derde geval 14 pens IC's in plaats van 16 pens, wat weer ruimte op de print scheelt die, zoals de pessimist Murphy reeds voorspelde, bij het vorderen van het ontwerp altijd net te klein blijkt te zijn. In mijn shack klopt noch de Wet van Murphy noch de Wet van Ohm. Het is maar dat je het weet.

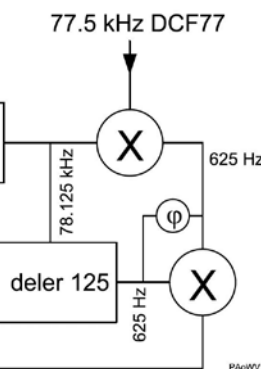


Fig. 5

Als ik (zie het blokschema in **figuur 1**) de 10 MHz-oscillator door 125 deel, houd ik 80 kHz over. Die 80 kHz meng ik met de ijkzenderinput van 77,5 kHz. Dat levert dan 2,5 kHz verschilfrequentie. Tevens deel ik de 80 kHz verder door 32, waardoor 2,5 kHz resteert. Die meng ik met de uitgang van de eerste mengtrap en houd dan 0 Hz over, oftewel een gelijkstroomcomponent. De grootte daarvan is afhankelijk van de faseverschuiving van de TCXO ten opzichte van de ijkzenderinput, en die wordt gebruikt om de TCXO bij te regelen tot zijn frequentie en fase gelijk zijn aan het gemiddelde van DCF77.

Une autre solution est donnée à la **figure 2**. Elle consiste à diviser la fréquence du TCXO par 125, ce qui donne 80 kHz. On divise ensuite par 32 pour obtenir 2,5 kHz. On mélange alors le 80 kHz avec le 2,5 kHz pour obtenir 77,5 kHz par différence. Cette dernière fréquence peut alors être comparée à la fréquence de référence de DCF77 pour produire le signal d'erreur.

Supposons par exemple que dans le schéma bloc de la **figure 1**, le TCXO soit 1 Hz trop haut. Cela correspond à 1/125 de Hz trop haut à 80 kHz et 1/4000 de Hz à 2,5 kHz. La sortie du premier mélangeur vaut donc 2,5 kHz + 1/125 Hz et celle du deuxième mélangeur 1/125 - 1/4000 Hz, soit 0,00775 Hz. Cela signifie qu'un phasemètre raccordé là (par exemple le Zerobeater décrit dans le n° 10 de CQ-PA 2009) a besoin de deux bonnes minutes pour parcourir 360°. On a donc un écart de fréquence de 1 sur 10^7 , clairement mesurable sur un phasemètre. Pour une précision de 1 sur 10^9 , une révolution complète de 360° de la composante continue du second mélangeur dure 100 fois plus longtemps, soit plus de trois heures et demie. Une telle précision (une erreur de 1 Hz sur 1 GHz) ne pourra être atteinte, car, à cause de la gigue de phase et des différences de chemins parcourus, l'émetteur étalon n'est pas suffisamment stable à court terme; sa fréquence doit alors être moyennée sur une plus longue période pour obtenir la précision désirée. En fait, cela signifie que notre propre standard de fréquence doit présenter, en boucle ouverte, cette précision sur la période indiquée ci-dessus, parce qu'autrement, il dériverait trop pendant la longue période de moyennage. Une précision de 1 Hz sur 100 MHz suffit pour l'objectif annoncé, mais à cause de toutes sortes d'influences perturbatrices, on obtient toujours moins que ce que l'on veut, et pour cette raison, je m'efforce d'obtenir 1 sur 10^9 , autrement dit 1 Hz sur 1 GHz. Cela correspond pour une horloge à une erreur d'une seconde sur 31 ans. Cela signifie donc, comme nous allons le voir, que la fréquence de l'émetteur étalon doit être moyennée sur une période de 600 secondes. Pendant ce temps, le TCXO ne doit pas s'écarter de plus de 0,01 Hz de sa valeur nominale de 10 MHz.

Gigue de phase de l'émetteur étalon et précision disponible

La gigue de phase de DCF77 mesurée comme variation de la valeur nominale du temps de passage par zéro, vaut environ 0,6 microseconde en Hollande; avec cet ordre de grandeur, elle se situe donc dans le signal reçu. Donc sur 77,5 kHz, car à cette fréquence, une période dure 12,9 microsecondes, presque 17°. De plus, nous avons une modulation de phase, avec la garantie que les déviations sont en nombre pair et donc s'éliminent.

Si l'on veut avoir une précision de 1 sur 10^9 à partir de ce signal, il faut prendre au moins $10^9/21$ périodes du signal DCF comme temps de mesure (et aussi un nombre pair), parce qu'à cause de la gigue, on peut gagner ou perdre 17 degrés – c'est à dire 1/21 de période – pendant le temps de mesure, c'est-à-dire 600 bonnes secondes à 77,5 kHz. Pour atteindre une précision de $1E13$ sur la fréquence de référence, il faudrait un temps de mesure 10000 fois plus long, soit 71 jours.

Revenons à notre objectif, une précision de $1E9$. Si votre TCXO dérive de plus de 1 sur 10^9 (1/100 Hz) en 10 minutes, vous ne pouvez pas atteindre cette précision. En effet, si vous prenez des temps de mesure plus courts pour corriger plus souvent le TCXO, cette correction n'est pas fiable à cause de ce temps de mesure trop court pour la précision désirée. Il est aussi possible d'utiliser une moyenne mobile.

Le zerobeater n'est pas conçu pour comparer de tels signaux de fréquences extrêmement basses. La fréquence d'entrée la plus basse, à laquelle sa précision est aussi la plus élevée, se situe aux alentours de 30 Hz.

Considérons le premier schéma bloc (**fig.1**) et raccordons le zerobeater à l'entrée 2,5 kHz du deuxième étage mélangeur. Dans ce cas, le zerobeater convient bien pour mesurer et enregistrer ces différences de phase. Sur le schéma bloc de la **fig. 2**, dans le cas d'un écart de 1 Hz du TCXO, il faut alors de nouveau 2 bonnes minutes pour avoir une variation de 360°

Je kunt het ook anders doen, zoals **figuur 2** laat zien, namelijk de TCXO delen door 125, dat geeft 80 kHz, die frequentie doordelen met een 32-deler tot 2,5 kHz. Die 2,5 kHz en 80 kHz samen mengen levert 77,5 kHz verschilfrequentie en die mengen met de ijkzenderinput geeft ook een DC component als regelsignaal.

Stel dat in het blokschema in **figuur 1** de TCXO er om te beginnen 1 Hz te hoog naast staat. Dat is dan op 80 kHz 1/125 Hz te hoog en op 2,5 kHz 1/4000 Hz te hoog. De output van de eerste mengtrap is dus 2,5 kHz + 1/125 Hz en de output van de tweede mengtrap is dan 1/125-1/4000 Hz te hoog. Dat is 0,00775 Hz te hoog. Dat wil zeggen dat een daar aangesloten fasemeter (b.v. de Zerobeater CQ-PA 2009 nr 10) er ruim 2 minuten over doet om een volledige cirkel van 360 graden te beschrijven. Dan heb je dus een frequentieafwijking van 1 op 10^7 die op een fasemeter duidelijk op het oog geregistreerd wordt. Wil je regelen op 1 op 10^9 , dan duurt een volledige faseomwenteling van de DC-component uit de tweede mixer 100 keer zo lang dus ruim 3,5 uur. Een dergelijke nauwkeurigheid (1 Hz fout op 1 GHz) zal wellicht niet haalbaar zijn, want door de fasejitter en padlengtevariëaties is de ijkzender op korte termijn niet stabiel genoeg en moet je uitmiddelen over een langere tijd om daaruit de juiste gemiddelde frequentie te kunnen vinden, die voldoende de gewenste nauwkeurigheid garandeert. In feite betekent dat, dat je eigen vrijlopende standaard in open loop die nauwkeurigheid moet tonen over die periode, omdat hij anders tijdens de lange middeling te ver wegloopt. Een nauwkeurigheid van 1 Hz op 100 MHz volstaat voor het geschetste doel, maar je krijgt altijd minder dan je wilt, door allerlei Zuster Buitenhuis effecten, en daarom streef ik naar 1 op 10^9 , of 1 Hz eraan op 1 GHz, en dat komt overeen, als je een klok erop aansluit, dat die per 31 jaar 1 seconde kan foutlopen. En dat betekent weer, zoals we zullen zien, dat de ijkzender zijn frequentie uitgemiddeld moet worden over een periode van 600 seconde, en de TCXO in die tijd dus niet meer spontaan mag weglopen dan 0,01 Hz op zijn nominale waarde van 10 MHz.

Fasejitter van de ijkzender en haalbare nauwkeurigheid

De fasejitter van DCF77 is, gemeten als variatie van de nominale nuldoorgangstijd, ongeveer 0,6 microseconde in Nederland; in die grootteorde ligt hij in het ontvangen signaal. Dat is dan op 77,5 kHz, omdat een periode daar 12,9 microseconde duurt, bijna 17 graden. Bovendien wordt er fasegemoduleerd, met de garantie dat het aantal faseafwijkingen even is en elkaar dus opheft, dat is er later "na mijn tijd" bijgekomen. Ze blijven wijzigen, want de eerste 20 bits van een frame waren grotendeels ongebruikt en daar wil men nu een soort BB-alarm mee realiseren.

Wil je dus 1 op 10^9 nauwkeurigheid van dat signaal hebben, dan moet je minstens $10^9 / 21$ perioden van het DCF-signaal als meetduur aannemen (en tevens een even aantal) omdat je door de jitter 17 graden – dat is 1/21 deel van een periode – extra of tekort kunt binnenkrijgen in die meettijd; en dat is bij 77,5 kHz ruim 600 seconde signaalduur. Zou je de zender nauwkeurigheid van $1E13$ willen hebben dan moet je nog eens 10000 keer zo lang meten en dat is 71 dagen.

Terug naar ons ontwerpdoel, het $1E9$ geval: verloopt je TCXO in 10 minuten meer dan 1 op 10^9 (1/100 Hz) als random verandering van drift, dan kun je die nauwkeurigheid dus niet halen want als je kortere tijd meet om de TCXO vaker te corrigeren is je correctie navenant onbetrouwbaarder door de te korte meetduur voor die nauwkeurigheid. Je kunt wel een lopend gemiddelde van die tijdsduur aanhouden. Dat helpt.

De zerobeater is niet gemaakt om dergelijke extreem laagfrequente signalen te vergelijken. De laagst inputfrequentie, waar hij ook met de hoogste resolutie meet, ligt in de buurt van 30 Hz.

Nemen we het eerste blokschema (**fig.1**) en plaatsen we de zerobeater op de 2,5 kHz inputsignalen van de tweede mengtrap, dan is de zerobeater wel geschikt om die faseverschillen te meten en te registreren. Bij het blokschema in **fig. 2** heeft bij 1 Hz afwijking van de TCXO ook

sur la phase de la tension de réglage. Il n'est cependant pas possible de trouver dans ce schéma, deux fréquences égales et supérieures à 30 Hz, dont nous pouvons enregistrer le déphasage mutuel avec le zero-beater.

Le zero-beater mesure le déphasage comme un rapport entre l'intervalle de temps entre deux passages par zéro positifs des signaux d'entrée f_0 et f_1 , et la durée de la période de ces signaux. Cette mesure se fait avec un compteur à 16 bits piloté par une horloge à 1843 kHz; de cette façon, 2,5 Hz correspond à 737 pas de comptage et le glissement de phase peut être mesuré et enregistré avec une résolution de $0,5^\circ$. Si nous prenons $m = 128$ et $n = 128 * 125 = 16000$ pour les diviseurs (fig. 5), cette fréquence vaut alors 625 Hz. Dans ce cas, la résolution de la mesure de phase est 4 fois plus grande, donc largement $0,1^\circ$ de différence de phase. La chaîne de diviseurs se compose alors de 4 CI à 14 broches.

Considérations sur la conception

Nous pouvons réaliser les diviseurs avec des 74HTC ou des 74LS. La préférence, si on a le choix, va à la série moderne 74HCT. L'étage mélangeur peut aussi être réalisé de manière digitale avec une bascule type D ou une porte OU exclusif. Cette dernière peut en effet être conçue comme un buffer ou un inverseur d'une branche d'entrée, tandis que l'autre branche d'entrée choisit si elle est buffer ou inverseur. En plaçant un bloc sur cette entrée, la phase de l'autre signal sera ou non renversée, c'est-à-dire multipliée par 1 ou -1 et on aura donc le mélange des deux fréquences d'entrée. Une bascule type D échantillonne le signal à l'entrée D, et multiplie donc avec la fréquence d'échantillonnage présente à l'entrée horloge; il y a aussi une fonction d'échantillonneur-bloqueur qui atténue les harmoniques de rang élevé.

A la sortie du premier étage mélangeur, qui mélange le 77,5 kHz avec le 80 kHz, on trouve non seulement du 625 Hz mais aussi, dans le voisinage immédiat, un tas d'autres composantes fréquentielles. La fréquence fondamentale (fréquence de répétition d'un motif d'impulsion) de ce signal binaire est de 625 Hz. Cependant, à cause des impulsions supplémentaires dues aux fréquences somme et différence ainsi qu'aux fréquences d'entrée, ce signal de sortie n'est pas facilement exploitable sans filtre passe bas à 625 Hz. C'est une soupe d'impulsions et lors du passage des flancs, des pics viennent aussi s'ajouter. Comme je ne désire pas placer un filtre analogique ou à capacités commutées à la sortie 625 Hz, j'ai décidé d'abandonner cette approche.

La réalisation digitale complète

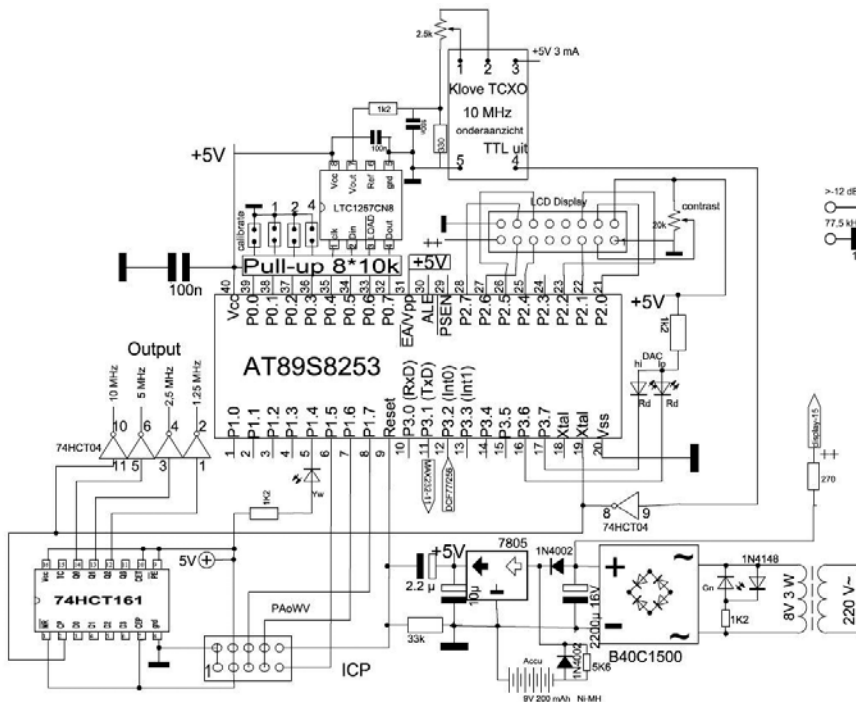


fig. 6 de processor en DAC

dan weer de fase van de regeltspanning 360 graden doorlopen in ruim 2 minuten, maar in dit schema zijn nergens twee gelijke frequenties groter dan 30 Hz te vinden die we onderling voor de registratie met de zero-beater de fase van kunnen registreren.

De zero-beater meet de fasehoek als een verhouding tussen de tijd tussen twee positieve nuldoorgangen van de inputsignalen f_0 en f_1 en de periodeduur van die signalen, met een 16-bitsteller die op een klok van 1843 kHz draait, zodat 2,5 kHz 737 counts omvat en de faseverschuiving dus met een resolutie van $0,5$ graad te meten en te registreren is. Nemen we het getallenpaar $m=128$ en $n=128*125=16000$ voor de delers (fig. 5) dan wordt die frequentie 625 Hz, zodat het voor de hand ligt om die te kiezen. Dan is de de resolutie van de fase meting 4 keer hoger, dus ruim $0,1$ graad faseverschil. De deelketen bestaat dan uit vier 14 pins IC's.

Overwegingen bij het ontwerp

We kunnen de delers met 74HTC of 74LS IC's realiseren. Voorkeur, als je kunt kiezen, zijn de modernere 74HCT-typen. De mengtrap kan ook digitaal worden uitgevoerd met een D-flipflop of een exclusieve-orpoort. Die laatste kan immers worden opgevat als een buffer of een inverter van een ingangspoot, terwijl de andere ingangspoot kiest of hij buffer of inverter is. Zet je een blok op die poot dan keert hij dus al of niet de fase van het andere signaal om, dat is vermenigvuldigen met 1 of -1 en dus mengen van de twee ingangsfrequenties. Een D-flop bemonstert het signaal op de D-input, en vermenigvuldigt dus met de bemonsterfrequentie, die op de klokinput staat, terwijl er tevens een sample-and-hold functie inzit die de hogere harmonischen tempert.

Uit de eerste mengtrap, die 77,5 kHz blok met 80 kHz blok mengt, komt niet alleen 625 Hz maar ook een hoop andere dicht in de buurt liggende componenten. Dat binaire signaal heeft als grondfrequentie (herhalingsfrequentie van een identiek pulspatroon) 625 Hz, maar door de extra pulsen die de som en verschilfrequentie en de inputfrequenties presenteren, is het zonder laagdoorlatend filter van 625 Hz niet makkelijk te gebruiken. Het is pulsoep en tijdens passeren van flanken komen er ook haren in die soep. Daarom heb ik, omdat ik geen analoog of switched capacitor filter in de 625 Hz output wil zetten van deze benadering afgezien.

De volledig digitale realisatie

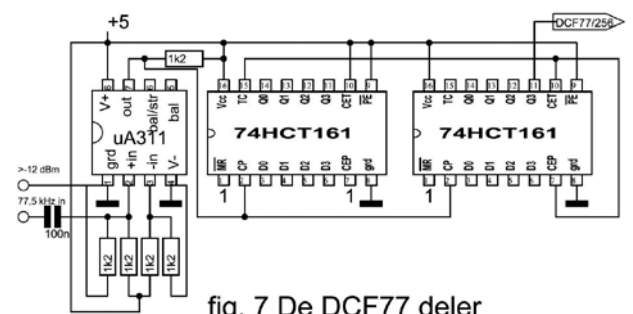


fig. 7 De DCF77 deler

Le signal DCF77 de 77,5 kHz passe à travers un limiteur et est ensuite utilisé comme impulsion d'interruption à l'entrée d'un microprocesseur. Pour ne pas surcharger le microprocesseur, une division du signal s'impose avant de l'envoyer à l'entrée interruption. Nous ne voulons pas non plus avoir trop d'impulsions les unes à la suite des autres; sur un intervalle de temps plus long, elles deviennent plus difficiles à distinguer. C'est comme lorsque l'on compte les pavés d'un trottoir, on se trompe facilement. Ce train d'impulsions à 77,5 kHz de DCF77 passe donc dans deux diviseurs par 16, ce qui donne une division par 256. Le contrôleur reçoit donc un signal d'interruptions dont la fréquence est de l'ordre de 300 Hz. Ce signal est alors facilement traitable et comme le diviseur est pair, nous sommes aussi quittes de la modulation de phase de DCF77.

J'emploie le contrôleur 89S8253 de Atmel car j'en possédais un exemplaire en attente d'une application. Ce contrôleur est piloté par le TCXO qui fournit un signal d'horloge de 10 MHz. Le contrôleur contient un diviseur fixe par 6 qui pilote un compteur interne à 16 bits (compteur n° 0). La période de ce compteur à 16 bits qui fonctionne en continu, correspond à une fréquence d'environ 25 Hz.

Considérons maintenant les interruptions DCF77 et regardons l'état du compteur 16 bits au moment des interruptions. Il apparaît alors que si les fréquences valent précisément 10 MHz et 77,5 kHz, les interruptions donnent chaque fois une autre valeur du compteur. On peut cependant calculer que si l'on laisse tourner le compteur 125 fois, (cela dure donc presque 5 secondes), le compteur affichera à la 125^{ème} fois la même valeur lors de l'arrivée de la 1488^{ème} interruption en provenance du diviseur DCF, pour autant que les fréquences du TCXO et de DCF77 n'aient pas changé entretemps.

Le calcul se fait comme suit:

$77,5 \text{ kHz} = 2^2 \times 5^4 \times 31$. La période vaut donc $1/(2^2 \times 5^4 \times 31)$. La fréquence du TCXO est de 10 MHz, soit $2^7 \times 5^7$, celle-ci étant divisée par 6 dans le contrôleur. L'horloge du compteur à 16 bits a donc une période de $3 / (2^6 \times 5^7)$ ou 0,6 microseconde. C'est la même valeur que la gigue de phase du signal de DCF77, donc, la division est suffisante. La période du compteur à 16 bits dans le contrôleur vaut donc $3 \times 2^{10} / 5^7$. Ce qui correspond à une fréquence d'environ 25 Hz. Ce compteur repasse donc de son maximum à zéro 25 fois par seconde.

Si nous voulons maintenant qu'après n boucles de ce compteur, une impulsion DCF d'interruption intervienne pour le même état du compteur, les nombres entiers m et n doivent satisfaire à la condition: m fois la période des impulsions DCF77 d'interruption = n fois la période du compteur. Ce qui donne:

$$m \times (1/(2^2 \times 5^4 \times 31)) = n \times 3 \times 2^{10} / 5^7 \text{ pour } m \text{ et } n \text{ entiers,}$$

$$\text{donc: } m/n = (3 \times 2^{12} \times 31) / 5^3$$

Nous voyons donc qu'après $5^3 = 125$ périodes du compteur, c'est-à-dire 5 secondes, les impulsions DCF tombent sur le même état du compteur. Les impulsions DCF précédente et suivante tombent alors 21 pas plus haut et 21 pas plus bas que s'il n'y avait pas de prédivision par 256 et que le signal DCF n'était pas entaché d'une modulation de phase.

Il n'est pas souhaitable que tant d'interruptions (77500) arrivent par seconde au processeur qui, s'il en manque une, tire des conclusions erronées. Cela explique la division externe du signal DCF77 par $2^8 = 256$ (avec un diviseur par 16 en plus, on pourrait aussi diviser par 2^{12}). Il y a donc dans les 5 secondes que le compteur tourne 125 fois, $2^4 \times 3 \times 31 = 1488$ interruptions. Dans le compteur, les états sont séparés l'un de l'autre de 5505 par interruption DCF.

On peut aussi appliquer ce calcul sur la porteuse de Droitwich (198 kHz), on obtient alors comme résultat, avec un prédiviseur par 256, que le compteur se retrouve dans le même état après 625 tours. Le diviseur par 256 a alors fourni 19008 impulsions d'interruption et la période d'échantillonnage vaut presque 25 secondes.

Het DCF77 signaal van 77,5 kHz gaat door een limiter, en wordt vervolgens gebruikt als interruptpuls op de ingang van een microprocessor. Om de processor niet te overbelasten is een deling van het signaal gewenst, voor we het aan de interruptpinnen aanbieden. We willen trouwens ook niet teveel pulsen achter elkaar; ze worden dan over een langere tijd moeilijker van elkaar te onderscheiden. Het is alsof je stoeptegels op het oog telt, je mistelt je dan makkelijk. Dus wordt die DCF77 77,5 kHz pulsreeks gedeeld door een getal waar ik twee 16 delers voor neem, zodat gedeeld wordt door 256. Dat levert dan een interruptsignaal van ruim 300 Hz aan de controller. Dat is dan wel te behappen, en het deeltal is even zodat we tevens van de fasemodulatie op DCF77 af zijn.

Ik gebruik een controller die ik heb, de Atmel 89S8253, omdat die hier op de plank ligt te wachten op toepassing. Die controller stuur ik met de TCXO als klok, 10 MHz dus. In de controller zit een vaste zedeler en die stuurt dan een 16 bits inwendig aanwezige teller_0 aan. De periodeduur van die 16-bitsteller die alsmaar doorloopt, komt overeen met ruim 25 Hz.

Geef ik nu met DCF77 interrupts en kijk ik op het moment van de interrupt naar de stand van de 16-bits teller, dan blijkt dat, als de frequenties 10 MHz en 77,5 kHz precies op waarde staan, die interrupt iedere keer een andere waarde te geven van de teller. Je kunt echter berekenen dat als je de teller 125 keer door laat draaien (dat duurt dus nagenoeg 5 seconde), dat die 125^{ste} keer de teller weer precies dezelfde waarde aangeeft op de inmiddels 1488^{ste} interrupt uit de DCF-deler, als de frequenties van TCXO en DCF77 onderling niet verlopen zijn.

De berekening luidt als volgt:

$77,5 \text{ kHz} = 2^2 \times 5^4 \times 31$. De periodeduur is dus $1/(2^2 \times 5^4 \times 31)$. De TCXO is 10 MHz, dat is $2^7 \times 5^7$, die in de controller wordt gedeeld door 6. De klok van de 16-bitsteller heeft dus een periodeduur van $3 / (2^6 \times 5^7)$ of 0,6 microseconde. Dat is dezelfde waarde als de fasejitter op DCF77, dus fijn genoeg verdeeld. De periodeduur van de 16 bits teller in de controller is dus $3 \times 2^{10} / 5^7$. Dat is ongeveer 25 Hz. Die teller draait dus 25 keer per seconde door zijn maximum heen terug naar 0.

Willen we nu dat een DCF-interruptpuls na n keren doordraaien van die teller op dezelfde tellerstand optreedt, dan moet met m en n gehele getallen gelden dat m maal de periodeduur van DCF77-interrupts gelijk moet zijn aan n keer de tellerperiode:

$$m \times (1/(2^2 \times 5^4 \times 31)) = n \times 3 \times 2^{10} / 5^7 \text{ voor } m \text{ en } n \text{ geheel,}$$

$$\text{dus: } m/n = (3 \times 2^{12} \times 31) / 5^3$$

We zien dus dat na $5^3 = 125$ tellerperioden, wat neerkomt op 5 seconde, de DCF-puls op dezelfde tellerstand valt. De vorige en volgende DCF-puls vallen dan 21 counts hoger en 21 lager als er geen voordeling door 256 plaats zou vinden en er geen fasemodulatie op het DCF-signaal zou zitten.

Het is ongewenst dat er zoveel (77500) interrupts per seconde aan de processor gegeven worden die dan, als er een ontbreekt, foute conclusies trekt. Vandaar dat we mede daarom een externe deling op het DCF-signaal doen van, zoals nu blijkt, een toegelaten factor van $2^8 = 256$ (nog een 16-deler erbij en dus 2^{12} zou ook mogen). Dan zitten er in de pakweg 5 seconde dat de teller 125 keer rondloopt $2^4 \times 3 \times 31 = 1488$ interrupts. In de teller liggen de tellerstanden per DCF-interrupt dan 5505 uit elkaar.

Je kunt deze berekening ook op de draaggolf van Droitwich (198 kHz) loslaten, dan krijg je met een 256-voordeler als resultaat dat de teller in dezelfde stand staat na 625 keer ronddraaien. Er zijn dan 19008 interruptpulsen uit de 256-deler gekomen van Droitwich, en de be-monsterperiodeduur wordt bijna 25 seconde.

Wat we nu gaan doen is het volgende. We laten een 16-bitsteller lopen, die elke keer als hij van maximum naar 0 doordraait een overflow-interrupt int1 aanroep. Die interruptafhandelingsroutine telt het aantal keren, dat dat gebeurt, met 125 als een bijzonder getal.

Voici maintenant ce que nous allons faire. Nous laissons tourner un compteur à 16 bits. Chaque fois qu'il revient à zéro, il envoie une requête d'interruption dont la routine de traitement compte le nombre de fois que ce retour à zéro se produit, avec 125 comme nombre particulier.

En outre, il y a une interruption externe (int0) dont la routine de traitement est appelée par le signal DCF77 divisé par 256, c'est-à-dire plus de 300 fois par seconde. Cette routine de traitement d'interruption compte le nombre d'interruptions et, lorsque ce nombre atteint 1488 et que le compteur local de passage se trouve sur 125, copie et conserve l'état du compteur à 16 bits; les compteurs par 1488 et par 125 sont alors remis à zéro. L'état du compteur à 16 bits est inchangé. Lire l'état d'un compteur à 16 bits n'est pas si simple car celui-ci se compose de deux bytes et pendant que l'on lit un des bytes, l'autre peut changer. Pour la lecture, le compteur est donc bloqué et un nombre correspondant au nombre d'impulsions manquées est ajouté pour éviter un allongement de la période du compteur.

Il serait ennuyeux que le compteur 16 bits passe juste à son maximum lors de cette 1488^{ème} interruption externe, car il y aurait alors deux interruptions arrivant simultanément. Pour cette raison, lors de la mise sous tension, au moment de la première interruption externe, ce compteur est fixé à mi parcours et les compteurs de dépassement et d'interruptions externes sont mis à zéro. Cette procédure d'initialisation se répète lors d'une coupure du signal DCF; cela est signalé par un message affiché sur un petit écran LCD. Si une seule période de DCF77 n'arrive pas au diviseur par 256, ou si du fait d'une impulsion parasite, ce diviseur reçoit une impulsion d'horloge en trop, cela signifie un déplacement de 21 pas pour le compteur après 5 secondes du cycle de mesure; cela peut se constater par comparaison avec le résultat précédent du cycle de mesure. Cela signifierait en effet que le TCXO aurait subi soudainement un saut de fréquence de l'ordre de 25 Hz depuis le cycle de mesure précédent, donc en 5 secondes.

Le programme principal compare l'état actuel du compteur avec celui que ce compteur avait 5 secondes auparavant. Si l'état du compteur diffère du précédent de 1, cela veut dire que dans le cas d'un émetteur étalon sans gigue de phase, l'oscillateur local s'est décalé de 6 périodes, et a donc dérivé d'environ 1 Hz. Cet écart de 1 est dû en fait à la gigue normale (0,6 microseconde) de l'émetteur étalon affecté de modulation de phase; on ne fait donc rien dans ce cas-ci et on regarde si l'écart augmente la fois suivante. La différence est alors corrigée en ajustant la tension de réglage de l'oscillateur par l'intermédiaire d'un DAC série piloté par le processeur, de manière à ce que l'oscillateur descende de 0,1 Hz. Si l'écart augmente, alors on descend encore de 0,1 Hz. Et ainsi de suite jusqu'à ce que les choses soient rentrées dans l'ordre. On a donc affaire ici à un système intégrateur; cela veut dire qu'il ne se contente pas uniquement d'avoir la fréquence exacte, mais aussi le nombre exact de périodes sur le long terme. Aussi longtemps que l'écart augmente, nous effectuons toutes les 5 secondes, une correction de fréquence de 0,1 Hz vers le bas. Lorsque la fréquence est revenue à sa valeur nominale, nous n'y touchons plus jusqu'à ce que le compteur ait retrouvé son état nominal; nous augmentons ensuite la fréquence de 0,1 Hz en espérant que les choses ne changent plus. Cet algorithme ne produit donc pas une correction proportionnelle à l'écart, ce qui donne lieu à de plus grandes variations de fréquence (avec comme objectif d'avoir une correction plus rapide), lesquelles sont indésirables.

Le filtre PLL digital

Si, avec l'algorithme décrit ci-dessous, nous pouvons maintenir l'écart inférieur ou égal à un pas du compteur en tenant compte de la gigue de DCF77, l'oscillateur dérive alors de moins de 6 Hz au cours d'une période de mesure de 5 secondes, ce qui correspond à 1 sur 10^7 . Pour atteindre une précision de 1 sur 10^9 , nous devons utiliser une période de mesure d'au moins 10 minutes et moyenner les mesures. Cela peut se faire avec une boucle à verrouillage de phase utilisant un filtre à retard de phase. Il s'agit en fait d'un filtre RC composé d'une résistance R_2 dans la branche horizontale et dans la branche verticale, d'une capacité C en série avec une autre résistance R_1 . Ce type de filtre offre, par rapport à un filtre RC simple, l'avantage de permettre un réglage de l'amortissement critique

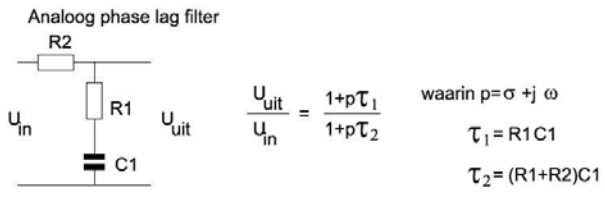
Voorts is er een externe interrupt met afhandelingsroutine int0 die op de door 256 gedeelde DCF77 wordt aangeroepen, ruim 300 keer per seconde dus. Die externe interruptafhandelingsroutine telt het aantal interrupts en laat het daar doorgaan bij. Is echter dat aantal 1488 geworden, en staat tevens (ter controle) de lokale doorgangsteller op 125, dan wordt de stand van de 16-bitsteller gekopieerd en bewaard en worden de 1488-teller evenals de 125-teller op 0 teruggezet. De stand van de 16-bitsteller wordt ongewijzigd gelaten. Zomaar een tellerstand uitlezen gaat niet, want die bestaat uit 2 bytes en als je de ene leest, kan de andere ondertussen veranderen. De teller wordt dus voor de uitlezing stilgezet, en omdat dat de tellerperiode zou verlengen wordt er bij de stilstaande teller ter correctie een bedrag opgeteld dat precies overeenkomt met de gemiste counts tijdens de uitschakeling.

Het zou onprettig zijn als de 16-bitsteller net ongeveer door maximum gaat op die 1488^{ste} externe interrupt omdat er dan twee interruptafhandelingen gelijktijdig om aandacht vragen. Daarom worden de eerste keer na inschakelen op het moment van de eerste externe interrupt die teller op halverwege zijn bereik gezet en de overflowteller en externe interruptteller op 0. Deze procedure, een initialisatie dus, herhaalt zich als het DCF-signaal een tijdje wegvalt (omdat de elektriciteitsmeter wordt uitgelezen door Big Brother, of de buurvrouw haar antieke TV inzet die blubberstoring maakt op veelvouden van 15625 Hz, dan wel in Braunschweig er door Oppergod Zeus niet gezwegen wordt maar uiting aan zijn toorn wordt gegeven met donder en bliksem). Dat wordt aan de buitenwereld gemeld op een LCD-schermpje. Als er maar een periode van het DCF77 signaal niet tot de 256 deler doordringt, of door een stoorpuls er een klokpuls teveel aan die deler gegeven wordt, betekent dit een verschuiving van 21 counts in de tellerstandwaarneming na 5 seconde meetcyclus. Dat is te herkennen, gelet op het bewaarde vorige resultaat van de meetcyclus. Het zou immers betekenen dat de TCXO ineens ruim 25 Hz in frequentie versprongen is sinds de vorige meetcyclus, dus in 5 seconde. Dergelijke resultaten gaan linea recta het vuilnisvat in (in 's-Gravenhage de asbak geheten, want Hagenezen produceren geen vuil – veronderstel zeg – die hebben uitsluitend as ten gevolge van het verbranden van vioolkisten voor het verwarmen van hun pied à terre au residence), oftewel die resultaten gaan in modern Nederlands de bitbucket in. Je spreekt op den duur de taal van de laatste overwinnaar. Toen Frans, nu Amerikaans-Engels. Elders Spaans.

Het hoofdprogramma bekijkt hoeveel de nieuwe tellerstand afwijkt vergeleken met die van 5 seconde geleden. Eerst op zijn JBF (JanBoerenFluitjes in kameelnotatie) als voorlopig gedachteproefje en om de werking te kunnen begrijpen, en daarna meer doordacht. Wijkt die tellerstand 1 af dan zou bij een ideale jittervrije ijkzender, de lokale kristaloscillator 6 perioden zijn verlopen, dus er ongeveer 1 Hz naast staan. Die ene count is echter de normale jitter (0,6 microseconde) in het ijkzendersignaal ontdaan van fasemodulatie, dus we doen niks, en kijken of het verloop toeneemt de volgende keer. Het verschil wordt dan gecorrigeerd door de regelspanning op de oscillator bij te stellen via een door de processor aangestuurde seriële DAC, zodanig dat de oscillator 0,1 Hz lager wordt gezet. Blijft er een toename, dan weer 0,1 Hz lager. Een en ander tot de zaak volledig is teruggelopen naar de beginstand. Het regelsysteem werkt dus integrerend, dat wil zeggen dat het niet tevreden is met een juiste frequentie, maar streeft naar het juiste aantal sinusperioden over lange termijn. Zolang de zaak omhoog loopt maken we dus per 5 seconde de frequentie 0,1 Hz lager. Loopt de frequentie terug naar nominaal dan laten we die constant tot 0 verschil met de nominale gewenste tellerstand bereikt is en verhogen hem dan 0,1 Hz in de hoop dat hij nominaal blijft hangen. Het JBF-algoritme is dus niet zo dat de correctie evenredig is met de fout, want dat geeft grotere afwijkingen in frequentie (met het doel die sneller te corrigeren) die ongewenst zijn. Leuk voor het begrip, maar eigenlijk knudde, net als die boerenfluit.

Het digitale PLL filter

Als we de zaak met een JBF-methode binnen 1 count kunnen houden die overeenkomt met de jitter van DCF77, dan verloopt de oscillator minder dan 6 Hz in de meetperiode van 5 seconde, dus dan staat hij binnen 1^9 . Om naar het streefdoel van nauwkeurigheid van 1^9 te



Digitale realisatie met IIR filter:
via Bilinear Tustin transform $p = \frac{2}{T} \frac{z-1}{z+1}$

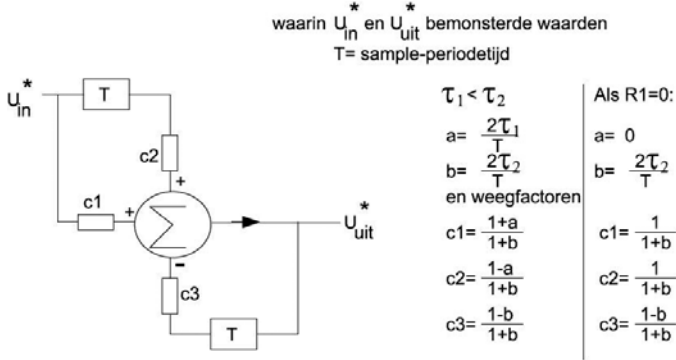


fig. 9 Phase lag PLL filter

indépendamment de la fréquence propre. L'amortissement du système est critique lorsque le temps de réponse du système est minimum, mais en restant à la limite du dépassement. La pulsation naturelle Ω_n de la boucle est la pulsation à laquelle il oscille lorsque l'amortissement est nul. Lorsque le facteur d'amortissement passe de 0 à 1, cette pulsation Ω_n tend vers 0. Les pôles, c'est-à-dire les racines du dénominateur de la fonction de transfert, se déplacent en effet en fonction du facteur d'amortissement, sur un demi cercle de rayon Ω_n . Lorsque le facteur d'amortissement vaut $\zeta=1$, l'amortissement de la boucle est critique. Avec un filtre RC simple, l'amortissement augmente de 12 dB par octave pour les hautes fréquences; en plaçant une résistance en série avec la capacité dans la branche verticale du filtre, on a alors un zéro dans le compteur et la pente est ramenée à 6 dB par octave. Le schéma d'un tel filtre est donné à la figure 9.

Nous désirons effectuer de manière numérique, le moyennage de la gigue sur ce long intervalle d'intégration de 10 minutes, avec comme entrée, l'état du compteur échantillonné toutes les 5 secondes. Cela peut se faire avec un filtre numérique de type IIR (voir la figure 9). Ce type de filtre a tendance à osciller à cause des erreurs d'arrondi. Les calculs devront donc être effectués avec une haute précision, c'est à dire avec beaucoup de bits.

La boucle à verrouillage de phase avec ce filtre peut être testée au moyen d'un échelon; c'est-à-dire en modifiant brusquement la fréquence de l'oscillateur à quartz (par exemple en modifiant le réglage du potentiomètre 10 tours du TCXO). On regarde alors comment varie la tension de réglage du TCXO pour ramener l'oscillateur à quartz à sa fréquence nominale. Cela doit se faire de manière assez rapide, sans dépassement de phase appréciable. Un dépassement en fréquence est nécessaire pour avoir un effet de rattrapage, car ce n'est pas tant la fréquence nominale qui compte pour l'indication à long terme de l'horloge, mais bien le nombre de sinusoïdes passées. La boucle fait donc en sorte qu'il n'y ait pas, avec l'amortissement critique, de dépassement de phase, mais pour atteindre ce but, il faut un dépassement en fréquence, car il faut rattraper ce que l'on a laissé en arrière en allant temporairement trop vite.

Les formules de conception de la PLL sont données à la figure 10. Le schéma bloc montre le principe de la PLL. La phase du TCXO est comparée à celle de l'émetteur étalon par le comparateur de phase qui fournit le signal d'erreur. Ce signal d'erreur est traité par le filtre à retard de phase pour donner le signal de commande qui est appliqué à la broche de commande du TCXO.

reiken, moeten we zoals gezien over ruim 10 minuten meten, en de metingen uitmiddelen. Dat kan met een phase locked loop waarin een zeer laagdoorlatend phase lag filter is opgenomen. In feite is dat een onderdoorlatend RC-filter bestaande uit een R in de langstak en een C in de dwarstak in serie met een andere R. Voordeel van dergelijk filter t.o.v. een enkel RC-lid is, dat je de kritische demping van de regelloop zo snel mogelijk de eindtoestand bereikt, maar nog net zonder overshoot. De natuurlijke frequentie Ω_n van de loop is de radiaal frequentie waarop hij oscilleert als de dempingsfactor 0 is. Bij het stijgen van de dempingsfactor tot 1 daalt die frequentie Ω_n tot 0. De polen, dat zijn de wortels van de noemer van de overdrachtsfunctie, bewegen namelijk als functie van de dempingsfactor over een halfcirkel met straal Ω_n . Als de dempingsfactor $\zeta=1$ is, is de loop kritisch gedempt. Heb je alleen één RC-lid dan neemt de demping voor hogere frequenties met 12 dB per octaaf toe; heb je echter een R in serie met de C in de dwarstak, dan heb je een nulpunt in de teller en wordt dat 6 dB per octaaf. Schema van zo'n netwerk staat in **figuur 9**.

Dat uitmiddelen van de jitter willen we over die lange integratietijd van 10 minuten digitaal uitvoeren op de om de 5 seconde gesampelde tellerstand als input. Dat kan met een digitaal filter worden gerealiseerd. We nemen dan een IIR-filter (zendexamenstof, hoef ik dus aan zendamateurs niet uit te leggen), zoals ook in **figuur 9** te zien is, en dat heeft de neiging door afrondingsfouten te gaan oscilleren. Daarom moeten de berekeningen met een hoge precisie, dus veel bits, worden gedaan omdat de oscillatieneigingen bij sommige inputpatronen ontstaan door afrondingsfouten in de berekeningen.

De phase locked loop met dat filter kan getest worden met een sprongverstoring, dus door plotsklaps de grondfrequentie van het kristal te

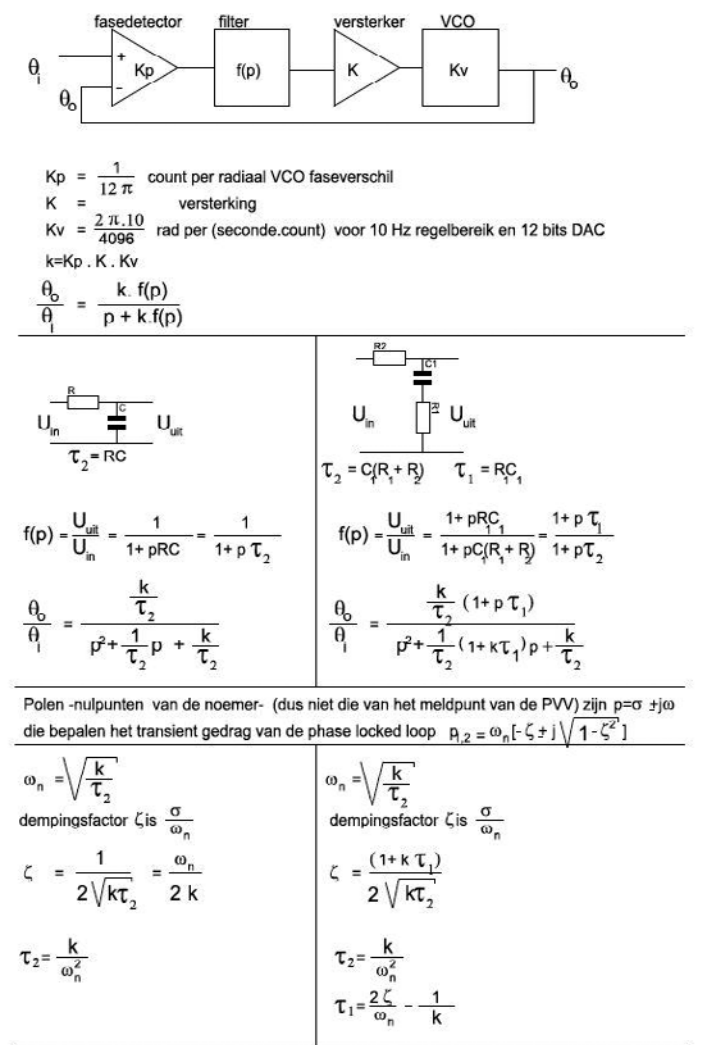


fig. 10 Linear PLL ontwerp

Ω_n est la pulsation naturelle de la boucle, et ζ le facteur d'amortissement qui doit être égal à 1 pour un amortissement critique. Il n'y a alors pas de dépassement et le temps de réponse est minimum. En posant τ , égal à zéro, on a affaire à un filtre RC simple (**figure 9**). Nous convertissons ensuite le filtre à retard de phase analogique en un filtre à retard de phase numérique; la représentation de la figure 9 est valable pour le signal d'entrée échantillonné ($T=5$ seconde). Le détecteur de phase dont nous avons parlé plus haut fournit 1 pas de différence pour 17° de différence de phase du passage par zéro de l'émetteur étalon après T seconde et aussi pour $12 \times \pi$ radians du TCXO. Cela est désigné par K_p dans les formules, K_p est exprimé en pas par radian de différence de phase sur 10 MHz entre DCF77 et le TCXO. Avec les valeurs de résistances données dans le schéma de la **figure 6**, le TCXO a une plage de réglage de 10 Hz lorsque le DAC varie entre 0 et 4095, soit $10 / 4096 = 0,00244$ Hz de variation de fréquence par pas du DAC, ou $0,00244 \times 2\pi$ radian par pas du DAC [count.seconde]. Ce coefficient est appelé K_v . Les coefficients c_1 , c_2 et c_3 sont déterminés à l'aide de la transformée en z , une sorte de transformée de Fourier pour signaux échantillonnés (voir la **figure 9**); La transformée de Tustin dont il est question à la figure 9 est une approximation du premier ordre de la transformée en z .

Lorsque l'on échantillonne, le spectre du signal échantillonné se répète indéfiniment. Par une transformation de l'axe des fréquences en un cercle (transformée en z), toutes ces répétitions tombent l'une sur l'autre sur le cercle; les points situés à gauche de l'axe des valeurs de p (p négatif) correspondent aux points à l'intérieur du cercle.

Etant donné que le temps de moyennage du filtre doit varier en fonction de la stabilité de l'oscillateur, les paramètres du filtre c_1 à c_3 sont calculés pour différentes valeurs de Ω_n et leurs valeurs mémorisées dans le contrôleur. Elles sont sélectionnables via 3 cavaliers sur les broches 38,37 et 36 du port P0, ces cavaliers sont désignés par les chiffres 1, 2 et 4 sur le schéma. Il est donc possible de sélectionner 8 filtres différents avec des temps de moyennage croissant. Par exemple, en plaçant des cavaliers sur 1 et 4, on obtient le filtre 5. Si aucun cavalier n'est placé, on a le filtre 0. Les valeurs des paramètres de ces 8 filtres sont données dans le tableau de la **figure 10**.

Le traitement des valeurs échantillonnées dans le filtre doit se faire avec une grande précision. A cet effet, on utilise ici un calcul en virgule flottante avec mantisse de 32 bits. L'avantage est que la transformation du binaire en BCD et inversement n'est pas nécessaire. Pour les besoins du YALC (CQPA nr 7/8 2009), j'avais développé à l'époque un programme de calcul en virgule flottante pour ce type de processeur et cela vient maintenant à point. Le résultat est ensuite converti en 12 bits pour le pilotage du DAC.

Comme les coefficients c_1 à c_3 ont tous au dénominateur le terme $1+b$, ce dénominateur pourrait être mis sous forme de puissance de 2, ce qui accélère la division; il suffirait ensuite de corriger le résultat au moyen d'un coefficient multiplicateur fixe. Cela est en fait tout à fait superflu car les coefficients c_1 à c_3 sont au préalable calculés avec un programme en C double precision et mémorisés dans la mémoire programme du contrôleur sous la forme de 8 séries de 3 constantes fixes.

La routine de division est exclusivement nécessaire lors du changement des cavaliers, pour conserver les mêmes conditions initiales pour le nouveau filtre, de façon à ce que la transition vers le nouveau filtre se fasse sans saut dans la sortie Y_n . La relation entre la sortie du filtre s'écrit: $c_1 * X_n + c_2 * X_{n-1} - c_3 * Y_{n-1} = Y_n$, X_n étant le $n^{\text{ème}}$ échantillon (voir dessin de la figure 9). Si l'on change brusquement les paramètres du filtre c_1 , c_2 et c_3 (en changeant les cavaliers), X_n , X_{n-1} et Y_n sont inchangés et, pour assurer la continuité de la sortie, Y_{n-1} doit donc être déterminé à partir des nouvelles valeurs de c_1 , c_2 et c_3 . Cela se produit à chaque changement des cavaliers à la volée.

C'est évidemment ennuyeux, lorsque l'on met en service un filtre lent, qu'il faille attendre des heures avant que le système soit stabilisé. Pour cette raison, on commence avec un filtre à réponse rapide (par un choix de cavalier), et, lorsque la réponse est stabilisée, on commute sur le filtre suivant et ainsi de suite, pour arriver plus rapidement à la stabilisation.

wijzigen (bijvoorbeeld door een zwengel aan de 10 turn instelpot van de TCXO te geven) en te kijken hoe de regelspanning van de TCXO zich dan gedraagt om dat kristal weer naar de nominale frequentie te vertrekken. Dat moet relatief snel naar de nieuwe eindwaarde kruipen zonder noemenswaardige overshoot in fase. Overshoot in frequentie is noodzakelijk als inhaaleffect, omdat niet de nominale frequentie maar het nominaal aantal verlopen sinussen van belang is voor de langetermijn klokaanduiding. De lus zorgt dus met kritische demping dat er geen overshoot is in de fase, maar om dat te bereiken is die er wel in de frequentie, want je moet inhalen wat je achterliep door tijdelijk sneller te lopen.

De ontwerpformules van de PLL staan in de tekening van **figuur 10**. We gaan bij het ontwerp uit van een lineaire analoge loop, dus fasevergelijker tussen de externe ijkzender, en de TCXO output, dat door het phase lag of RC-filter heen en de uitgang daarvan naar de regelpen van de TCXO.

Ω_n is de eigenfrequentie van de loop, en ζ is de dempingsfactor die voor kritische demping 1 moet zijn. Dan is er geen overshoot en gaat de fase zo snel mogelijk naar de eindwaarde toe. Door τ , nul te stellen gaat het phase lag netwerk over in een gewoon RC-netwerk, zoals uit **figuur 9** blijkt. Vervolgens zetten we het analoge phase lag filter om in de digitale representatie die voor bemonsterde ($T=5$ seconde) inputsignalen geldt. De fasedetector die we besproken hebben levert 1 count verschil voor 17 graden faseverschil van de ijkzender nuldoorgang na T seconde en ook voor $12 \times \pi$ radialen van de TCXO. Dat wordt in de formules K_p genoemd, uitgedrukt in count per radiaal faseverschil tussen de op 10 MHz gerelateerde fase van DCF77 en de TCXO. De TCXO heeft, met de in het schema in **figuur 6** aangegeven waarden van de weerstanden, 10 Hz regelbereik als de DAC tussen 0 en 4095 loopt. Dat geeft per count van de DAC inputstapje dus $10 / 4096 = 0,00244$ Hz verschil in frequentie, dat is $0,00244 \times 2\pi$ radialen per DAC [count.seconde]. Die noemen we K_v . Met de Z-transformatie, soort Fouriertransformatie voor bemonsterde signalen, helemaal het bittere einde want verder dan Z kun je niet gaan, worden dan de weegfactoren c_1 , c_2 en c_3 bepaald, zie de aanwijzingen in **figuur 9**. De daar genoemde Tustin-transformatie is een eerste orde benadering.

Als je bemonstert, repeteert het spectrum van het originele signaal dat je bemonstert steeds, en door transformatie van de frequentie-as op een cirkel vallen al die repetities over elkaar heen op de cirkel, vandaar die Z-transformatie. Het hele linker p-halfvlak ligt dan binnen de cirkel getransformeerd.

Omdat het filter ook voor stabielere of minder stabiele oscillatoren kleinere en grotere uitmiddelingstijd moet kunnen hebben, zijn de filterparameters c_1 t/m c_3 voor verschillende Ω_n berekend en opgenomen in de controller. Ze zijn kiesbaar middels 3 jumpers op port P0 op de IC-pennen 38, 37 en 36, in het schema aangegeven met 1, 2, 4. Daarmee zijn 8 verschillende filters instelbaar met toenemende uitmiddelingstijd. Een jumper op 1 en 4 betekent dus filter 5. Geen jumper geplaatst is dus filter 0. De filterparameters zijn voor de 8 filters in een tabel in **figuur 10** opgenomen.

Er moet met hoge nauwkeurigheid worden vermenigvuldigd, afgetrokken en opgeteld bij de verwerking van de monsterwaarden in het filter. Anders krijg je limit cycles, soort tinitus. Daarom is gebruik gemaakt van 32 bits mantissa floating point berekeningen. Voordeel is dat het meeste werk, namelijk omzetten van binair naar decimaal (BCD) en terug niet nodig is. Destijds heb ik een floating point pakket ontwikkeld voor dit type processor ten behoeve van de YALC (CQPA nr 7/8 2009) dat ik nu kon hergebruiken. Wie wat heeft, bewaart wat. Het rekenresultaat wordt in niveau verschoven en afgerond tot 12 bits voor de binaire DAC-sturing.

Omdat de filterfactoren c_1 t/m c_3 alle in de noemer de term $1+b$ bevatten zou die noemer een macht van 2 gemaakt kunnen worden wat de deling bespoedigt, en de output dan weer gecorrigeerd met een vaste vermenigvuldigingsfactor. Dat is echter volstrekt overbodig en dient geen enkel doel, omdat de coëfficiënten c_1 t/m c_3 tevoren met een C programma double precision zijn berekend en als 8 sets van 3 vaste constanten in het programmeergeheugen van de controller zijn opgenomen.

Lorsque le signal DCF est absent ou perturbé, de sorte que l'interruption n'arrive pas dans le domaine attendu, une LED jaune s'allume pendant un intervalle d'échantillonnage (5 s) et l'échantillon suivant à l'entrée du filtre est rendu égal au précédent. De plus, le nombre de fois que cela se produit est affiché sur l'écran LCD via un compteur de fautes.

Le DAC

Les mesures montrent que le TCXO régule la fréquence avec un tempo de 50 Hz par volt à sa broche de régulation. Cela signifie qu'avec une résolution de 1 sur 10^9 , on peut régler au maximum 1/100 Hz par pas du DAC; donc, 0,2 mV correspondent à 1 sur 10^9 . Un DAC à 8 bits doit alors être câblé de façon à ce qu'il couvre un domaine de 2,56 Hz maximum, donc 1,28 Hz plus haut ou plus bas que 10 MHz. C'est assez étroit et ne marche pas avec les filtres de n° inférieurs qui ont besoin d'un intervalle plus grand. C'est pour cette raison qu'un DAC à 12 bits à été choisi; avec la même résolution de 0,2 mV par pas, celui-ci permet un intervalle de réglage de presque 41 Hz.

Avec un DAC à 8 bits, l'écart maximum de l'oscillateur par rapport à sa valeur maximum est donc de 1,28 Hz. Cela correspond après T secondes = 125 tours du compteur, à une différence dans le comptage d'au moins 1 pas de comptage. Dans le cas d'un DAC à 12 bits, nous avons vu que l'intervalle de réglage maximum est 16 fois plus grand et que l'écart maximum par cycle de 125 tours de compteur est de 17 pas. Avec un DAC à 12 bits, nous pouvons diminuer l'intervalle de réglage en prenant une grandeur de pas plus petite que celle autorisée. Avec un diviseur de tension de 1k2 et 330 ohm, l'intervalle de réglage est de 10 Hz, ce qui est largement permis. Le facteur de conversion Kv entre la pulsation et l'entrée du DAC vaut alors au maximum $2 \cdot \pi \cdot 10 / 4096 = 0,015 \text{ rad}/(\text{seconde} \cdot \text{pas de comptage})$.

Explications supplémentaires

Les formules de calcul de c_1 , c_2 et c_3 de la **figure 6**, montrent que si l'entrée X_n est maintenue constante, et donc X_{n-1} aussi, Y_n et donc aussi Y_{n-1} sont constants et que l'on a donc $Y_n = X_n$, malgré tous les tracas de calcul en virgule flottante. Cela peut se calculer à partir de la formule du filtre $Y_n = c_1 \cdot X_n + c_2 \cdot X_{n-1} - c_3 \cdot Y_{n-1}$. En d'autres mots: le filtre a toujours pour chaque série de constantes calculée un gain de 1 pour la valeur de régime permanent. Juste comme un circuit RC dont il en est une traduction. Comme X_n est alors une constante entière, ce sera donc aussi le cas pour Y_n ; si l'on multiplie avec un facteur $K > 1$, on obtient comme régime permanent des pas du DAC K fois plus grands. En conclusion: la "granularité" du DAC est seulement complète si on l'utilise comme prévu, si $K \leq 4$, dans le cas où le DAC 12 bits a un intervalle de réglage de 10 Hz. En prenant $K = 40$ on perd donc 1 ordre et avec $K = 400$, 2 ordres, indépendamment du temps de moyennage du filtre pendant les changements de l'entrée de référence suite à la gigue.

Simulation

Si vous pensez que la théorie ne sert à rien parce qu'elle ne s'accorde pas à la pratique, vous pouvez naturellement essayer toutes sortes de choses jusqu'à ce que vous puissiez en tirer des conclusions, pour ensuite écrire sur des forums d'internet que vos résultats ont fait tomber des savants de leur chaise. Il est cependant beaucoup plus simple de simuler la PLL avec le filtre calculé sur un PC, avec quelques lignes de code en C ou en tout autre langage d'usage général avec lequel vous êtes habitués. Les échantillons sont pris non pas sur 5 secondes mais en quelques micro-secondes, et on voit tout de suite ce que cela donne à long terme. On peut alors ajouter de la gigue de phase avec un générateur de nombres aléatoires, l'objectif étant que si la variation moyenne due à la gigue est nulle, cela ne doit pas se retrouver dans la sortie pilotée par le DAC.

Vous pouvez donc tester si la gigue introduite aléatoirement sur DCF77 est suffisamment moyennée. Vous pouvez tester ce que vaut l'intervalle de capture de la PLL et comment vous devez choisir Ω_n pour moyennner la gigue de phase jusqu'à ce que la stabilité désirée pour le TCXO soit atteinte. Il y a aussi le fait qu'une boucle analogique fournit des nombres rationnels pour la tension de phase, mais en les convertissant en nombres naturels (entiers), on approche mieux la réalité parce que la phase est

De dealroutine is uitsluitend nodig om bij het wisselen van de jumpers dezelfde randvoorwaarden in te stellen voor het nieuwe filter zodat zonder sprong in de output Y_n continu wordt overgegaan op de nieuw ingestelde parameters. De verwerkingsformule van het filter is, als X_n het n-de monster is, zoals uit de tekening ook blijkt: $c_1 \cdot X_n + c_2 \cdot X_{n-1} - c_3 \cdot Y_{n-1} = Y_n$. Wijzigen plotsklaps door veranderen van de jumpers de filterparameters c_1 , c_2 en c_3 , dan zijn X_n , X_{n-1} en Y_n ongewijzigd, en moet dus Y_{n-1} voor de continuïteit van de uitgang uit deze filterformule worden bepaald op grond van de nieuwe waarden van c_1 , c_2 en c_3 . Dat gebeurt bij elke wijziging van de jumpers on the fly.

Nu is het vervelend, als je een traag filter instelt, dat je uren moet wachten tot de zaak de stabiele eindwaarde bereikt heeft. Daarom wordt bij instelling van een filter, door een jumperkeuze, begonnen met een lager geclassificeerd filter en als dat zijn eindwaarde snel heeft bereikt, overgeschakeld naar een volgend filter, tot uiteindelijk snel het met de jumpers ingestelde filter definitief wordt bereikt.

Als het DCF-sigitaal afwezig is of gestoord, zodat de interrupt niet in het verwachte gebied komt, gaat gedurende een monstertijd (5 s) een gele LED branden, en wordt het volgende inputmonster van het filter gelijk aan het voorgaande gemaakt. Voorts wordt het aantal keren dat dit gebeurd is, bijgehouden op het LCD-schermpje in een foutenteller.

De DAC

Bij meting blijkt dat de TCXO de frequentie regelt met een tempo van 50 Hz per volt op zijn regelpen. Dat betekent dat je bij 1 op 10^9 resolutie, maximum 1/100 Hz per stapje uit de DAC mag regelen en dus stapjes van 0,2 mV overeenkomen met 1 op 10^9 . Een 8-bits-DAC moet dan zo geschakeld worden dat die dan maximaal een bereik van 2,56 Hz over zijn volle bereik kan regelen, dus 1,28 Hz hoger of lager dan 10 MHz. Dat doet nogal krap aan, en loopt fout bij de lager genummerde filters die meer bereik nodig hebben. Daarom is een 12-bits-DAC gekozen die met dezelfde resolutie van 0,2 mV per stap een top-top regelbereik van bijna 41 Hz toestaat.

Bij een 8-bits-DAC is de maximaal instelbare afwijking van de oscillator uit de nominale waarde dus 1,28 Hz. Dat komt overeen met een verschil in de tellercount na T seconde = 125 tellerdoorlopen, van ruim 1 count. Bij een 12-bits-DAC is het maximale regelbereik zoals gezien, 16 maal groter en is de maximale afwijking per cyclus van 125 tellerdoorlopen, dus 17 counts. Met een 12 bits DAC kunnen we het regelbereik verkleinen door een kleinere dan de maximaal toelaatbare stapgrootte te nemen. Met een spanningsdeler 1k2 en 330 ohm is het top-top regelbereik dan 10 Hz wat nog ruim toelaatbaar is.

De conversiefactor Kv van DAC-input naar radiaal frequentie is dan maximaal $2 \cdot \pi \cdot 10 / 4096 = 0,015 \text{ rad}/(\text{seconde} \cdot \text{count})$

Groeiend inzicht

Als je de formules in **figuur 6** voor c_1 , c_2 en c_3 'aankijkt', zie je dat het filter, als de input X_n constant gehouden wordt en dus X_{n-1} ook constant is, dat uiteindelijk Y_n constant is en dus ook Y_{n-1} , en dat dan dus de waarde van $Y_n = X_n$, ondanks alle floating point gedoe. Dat kun je berekenen uit de filterformule $Y_n = c_1 \cdot X_n + c_2 \cdot X_{n-1} - c_3 \cdot Y_{n-1}$. Met andere woorden: het filter heeft altijd voor elke berekende set constanten c_1 , c_2 , c_3 een versterking 1 voor de steady state eindwaarde. Net als een RC-circuit, waar het een vertaling van is. Omdat X_n dan een constante integer is zal Y_n dat dus ook zijn; ga je dan vermenigvuldigen met $K > 1$, dan levert dat als steady state een veelvoud K van DAC-stappen op. Conclusie: de granulariteit van de DAC is slechts volledig als bedoeld gebruikt, als $K \leq 4$, in het geval de 12-bits-DAC 10 Hz regelspan heeft. Door $K=40$ te nemen verlies je dus 1 orde en $K=400$ 2 ordes, onafhankelijk van de uitmiddelingstijd van het filter tijdens wisselingen van de input tengevolge van jitter van de referentie-input.

Simulatie

Je kunt natuurlijk, omdat je van mening bent dat theorie nergens toe dient omdat die niet klopt met de praktijk, van alles gaan proberen en

K	filter	ω_n	sprongresponse op stap 10 uit fase-detector voor bereiken steady state analogoog		DAC-jitter bij random fasejitter uit fase-detector		Filter parms
			truncated	20000 samples	ampl 5, gem 0	ampl 50, gem 0	
100	0	5.75E-2	28 samples 1368 / 2101	27 samples 1412 / 2106	1607 / 2479	-2319 / 6326 out of range	c1=1.66667e-01 c2=1.66667e-01 c3=-6.66667e-01
$10\sqrt[3]{10}^2$	1	2.67E-2	63 samples 1740 / 2066	53 samples 1762 / 2069	1898 / 2198	507 / 3544	c1=8.49461e-02 c2=8.49461e-02 c3=-9.30108e-01
$10\sqrt[3]{10}$	2	1.24E-2	71 samples 1907 / 2055	101 samples 1915 / 2054	2006 / 2093	1618 / 2483	c1=4.13088e-02 c2=4.13088e-02 c3=-9.17382e-01
10	3	5.75E-3	153 samples 1983 / 2051	181 samples DACMIN=1987	2032 / 2063	1884 / 2197	c1=1.96078e-02 c2=1.96078e-02 c3=-9.60784e-01
$\sqrt[3]{10}^2$	4	2.67E-3	325 samples 2018 / 2049	352 samples 2020	2043 / 2053	1997 / 2100	c1=9.19779e-03 c2=9.19779e-03 c3=-9.81604e-01
$\sqrt[3]{10}$	5	1.24E-3	676 samples 2034 / 2048	681 samples 2035	2047 / 2049	2036 / 2061	c1=4.29038e-03 c2=4.29038e-03 c3=-9.91419e-01
1	6	5.75E-4	1360 samples DACMIN=2041	1336 samples DACMIN=2042	geen jitter	2045 / 2051	c1=1.99601e-03 c2=1.99601e-03 c3=-9.96008e-01
$\frac{1}{\sqrt[3]{10}}$	7	2.67E-4	2672 samples DACMIN=2045	2637 samples DACMIN=2045	geen jitter	2048 / 2049	c1=9.27457e-04 c2=9.27457e-04 c3=-9.98145e-01

Fig. 11 Filterparameters

tot Sint Juttemis wachten tot je conclusies kunt trekken, om dan vervolgens op internetfora te schrijven dat je geleerden van verbazing over je resultaten van hun stoel liet vallen. Het is echter veel eenvoudiger de PLL met het berekende filter te simuleren met een paar regels code geprogrammeerd in C of elke andere general purpose programmeertaal waar je vertrouwd mee bent, op een PC. De monsters worden dan niet per 5 seconde maar in enkele microseconden genomen, en je ziet direct aan de uitdraai wat er op lange termijn gebeurt. Je kunt dan fasejitter aanbrengen met een random number generator, en de bedoeling is dan, als het gemiddelde van de jitteruitwijkingen 0 is, dat je die niet terugvindt in de output die de DAC stuurt.

Je kunt dus testen of de random geïntroduceerde jitter op DCF77 voldoende uitmiddelt, je kunt testen wat het vangbereik is van de PLL en hoe je Ω_n moet kiezen om de fasejitter uit te middelen tot de gewenste stabiliteit van de TCXO bereikt wordt. Daar komt nog bij dat een analoge loop rationale getallen als fasespanning afgeeft, maar door die om te zetten in een natuurlijk (geheel) getal, je de werkelijkheid beter benadert, omdat immers de fase in gehele tellercounts gemeten en de TCXO in gehele DAC-stappen gestuurd wordt. Dat is dus een

toujours mesurée en pas de comptage et que le TCXO est piloté par des pas entiers du DAC. Il y a donc un effet de quantification qui aide le filtre qui travaille avec des échantillons (analogiques), et que l'on ne retrouve pas dans les formules car celles-ci sont faites pour des signaux analogiques.

Les résultats de cette simulation sont repris dans le tableau de la **figure 11**. La première colonne donne le gain de l'ampli entre le filtre et le DAC, afin de pouvoir choisir Ω_n . La deuxième colonne donne le n° du filtre que l'on peut mettre en service avec les cavaliers. La troisième colonne donne la valeur de Ω_n ; avec le filtre le plus lent, celle-ci est de l'ordre du dix millième de hertz, ce qui n'est pas réalisable avec un filtre analogique. Les 4^{ème} et 5^{ème} colonnes donnent le nombre d'échantillons nécessaires pour restabiliser la boucle suite à un saut de 10 pas dans le comptage de phase appliqué directement après le détecteur de phase. Ces valeurs sont données pour le modèle analogique et pour le modèle avec les valeurs arrondies à des nombres entiers (pas). Le filtre le plus lent exige 2637 échantillons, et cela dure 4 bonnes heures avant que le régime permanent soit de nouveau atteint; sous ces échantillons sont indiquées les valeurs prises par les niveaux de commande minimum et maximum du DAC. Avec les filtres les plus lents, ces excursions deviennent naturellement plus petites. Les deux colonnes suivantes indiquent la gigue du DAC suite à une gigue de phase aléatoire à la sortie du détecteur de phase, avec une amplitude de 5 et de 50 et une moyenne nulle. Le nombre d'échantillons est de 20000 dans chaque cas, ce qui représente une durée de plus d'un jour. De ces résultats, on peut conclure qu'une plus grande amplitude de gigue donne lieu à une gigue du DAC proportionnellement plus grande. Pour avoir une précision de 1E9, le DAC ne peut pas varier de plus de 8, suite à une gigue de 1 à la sortie du détecteur de phase (ce qui est en effet en accord avec la gigue de DCF77); cette condition peut être satisfaite avec le filtre 3 qui atteint sont état de régime permanent après 900 secondes (voir tableau), suite à un échelon de phase de 10 pas.

(à suivre)

N.d.l.r. Dans la littérature anglo-saxonne sur les PLL, le filtre à retard de phase de la **figure 9** est parfois aussi appelé "lag-lead filter". Dans ce type de filtre, contrairement au cas d'un filtre RC simple, le déphasage entre le signal de sortie et le signal d'entrée ne tend pas vers -90° lorsque la fréquence tend vers l'infini, mais passe par un maximum (en valeur absolue) pour ensuite revenir vers zéro lorsque la fréquence tend vers l'infini (donc: lag et puis lead). Cette appellation est cependant un peu trompeuse car "lag-lead filter" désigne aussi, dans la théorie du contrôle des systèmes, un filtre combinant les effets d'un filtre à retard de phase et d'un filtre à avance de phase.

quantiserings-effect waarmee het filter dat op bemonsterde (analoge monsters) signaal werkt, extra wordt behept en dat je niet in de formules terugvindt, omdat die voor analoge signalen zijn afgeleid.

Resultaten van deze simulaties zijn te vinden in de tabellen van **figuur 11**. De eerste kolom geeft de versterking aan die na het filter en voor de DAC wordt aangebracht, teneinde de Ω_n te kunnen kiezen. De tweede kolom geeft het filternummer, dat met de jumpers kan worden ingesteld. De derde kolom de Ω_n , die ligt dus bij het traagste filter in de orde van een tienduizendste hertz. Zoiets kun je analoog niet meer realiseren. Vervolgens is er een sprong in de fasecount direct na de fase-detector aangebracht van 10 counts en aan de hand daarvan is in de vierde en vijfde kolom berekend hoeveel samples (van elk 5 seconde tussentijd) nodig zijn om die te compenseren tot de zaak weer stabiel is. Het is bepaald voor het analoge model en voor het model met de waarden afgerond op gehele getallen (counts). Het traagste filter eist 2637 samples, en dat duurt dus pak weg 4 uur voordat dat weer steady state is. Onder die monsters staat vermeld wat de minimale en maximale uitsturing van de DAC is geweest. Die excursies worden met de tragere filters die er langer over doen uiteraard geringer. De volgende twee kolommen gaan uit van een stabiele toestand waar echter jitter aan de fase uit de detector wordt opgeteld ter grootte 5 en 50 maximaal, met een gemiddelde van 0. Bij elk vakje worden 20000 monsters genomen een tijdsduur vertegenwoordigend van langer dan een dag, en gekeken wat gedurende die meettijd de minimum- en maximumwaarde was die op de DAC optrad tengevolge van die jitter. Je kunt daaruit concluderen dat een grotere jitteramplitude is terug te vinden in een evenredig grotere DAC-jitter. Om 1E9 te halen mag de DAC niet meer variëren door jitter dan 8, bij een jitter uit de fase-detector van 1 (wat immers overeenkwam met de jitter van DCF77), zodat je met filter 3 kunt volstaan die na 900 seconde volgens de tabel zijn steady state bereikte bij een fasestapverstoring van 10 counts.

(wordt vervolgd)

N.v.d.r. Het filternetwerk gebruikt in de PLL (zie **figuur 9**) wordt in de Angelsaksische literatuur ook "lag-lead filter" genoemd. In tegenstelling tot een RC-filter, zal de faseverschuiving tussen het uitgangs- en het ingangssignaal niet naar -90° gaan wanneer de frequentie naar oneindig gaat, maar een absolute maximumwaarde bereiken en vervolgens naar nul gaan (dus: lag en vervolgens lead). De benaming is verwarrend omdat zij in de regeltheorie ook wordt gebruikt om een filter aan te duiden dat de effecten van een naïjlfiler en een voorjijlfiler combineert.