

TASC 2014-08 | The Quotient Transform (Ehlers)

Modified on 2014/06/27 08:05 by Eugene — Categorized as: TASC Traders Tips

Traders' Tip text

In this month's article, Mr. Ehlers introduces the **Quotient Transform** (QT for brevity) - another zero-lag filter that can be used for the purpose of timely trend detection. The QT is an advancement of the technique presented in January 2014 issue. Here, the output of a roofing filter (which includes applying a high pass filter and SuperSmoother filter), is normalized.

To execute the enclosed trading system, Wealth-Lab users need to install (or update to) the latest version of our **TASCIndicators** library from the **Extensions** section of our website if they haven't already done so, and restart Wealth-Lab.

A QT's drawback is that it's noticed to still stay in trend mode for too long after the uptrend is over. By applying two oscillators with different K parameters, author suggests that the system is facilitated to *exit or before the trend has run its course*. To demonstrate the application of the new oscillator, we followed author's example and used two QTs. The included trend-following system trades according to the rules below:

- Buy when the first 30-period QT with K = 0.8 crosses 0 from below
- Sell the position when the second 30-period QT with K = 0.4 crosses 0 from above

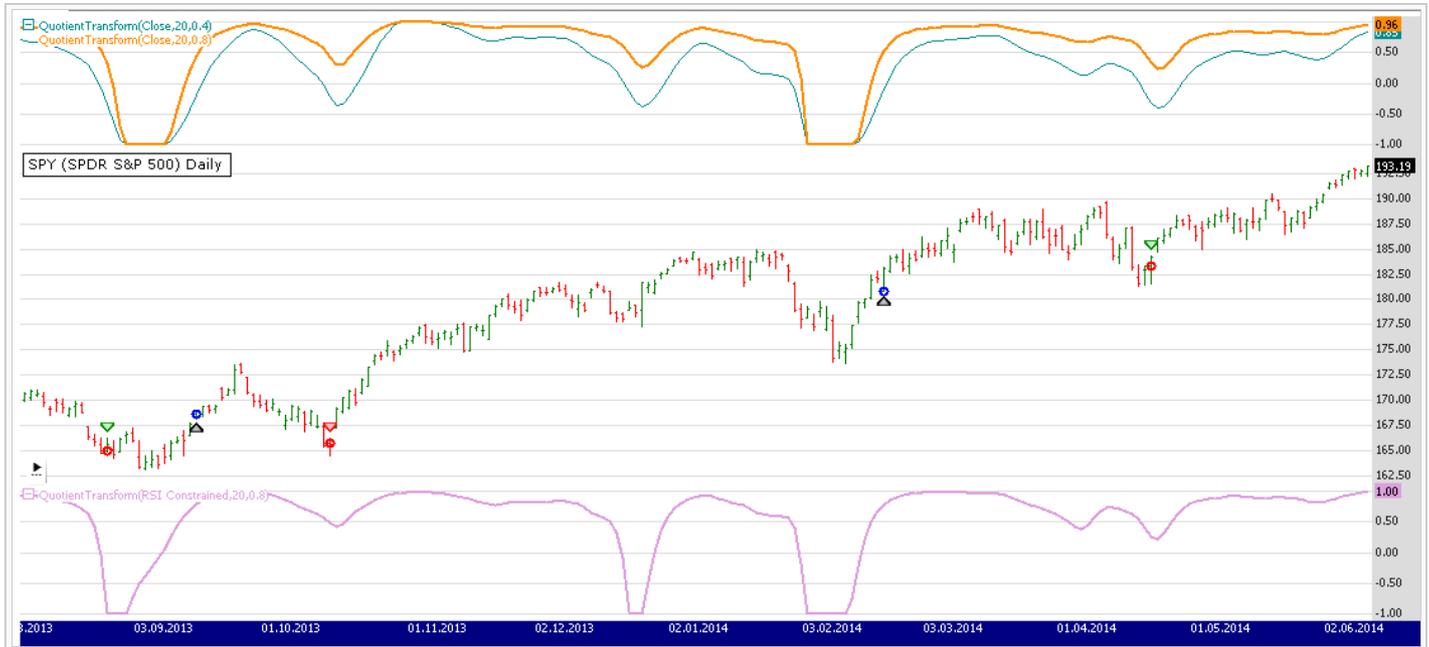


Figure 1. A Wealth-Lab 6 chart illustrating the application of the system's rules on a Daily chart of SPY. The upper pane plots two QTs with K = 0.8 and 0.4. The bottom pane shows a constrained RSI processed with Quotient Transform.

After applying the system to Dow 30 portfolio (10 years of Daily data, 10% equity per position, trading costs applied), we found that it had a drawdown too stressing and a market exposure too high to our taste, suggesting that a good portion of accumulated profits is still being given back to the market. However, that doesn't lessen the fact that in general, the system was successful (even though due to market's upside bias), beating the Buy&Hold's figure (115% vs. 103%) on 418 trades.

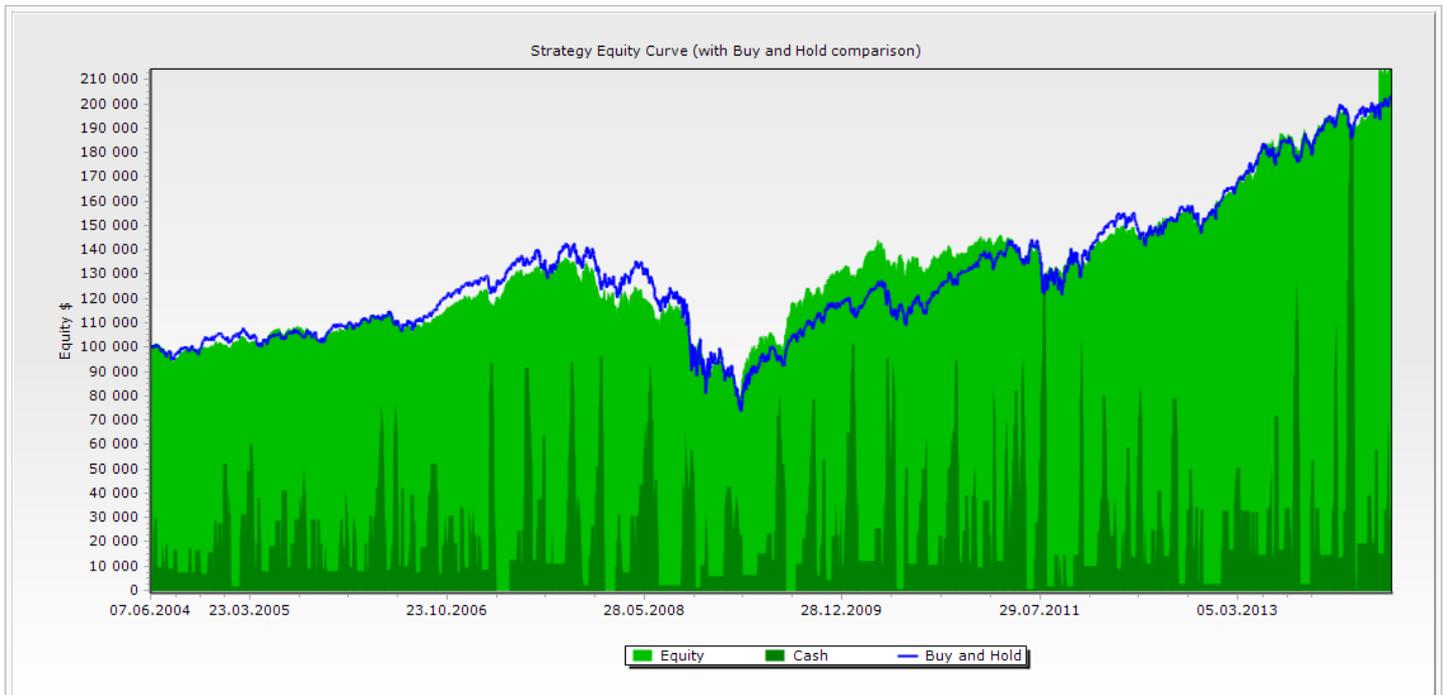


Figure 2. The system mostly followed the Buy&Hold.

Figure 1 points at another shortcoming of the simple system: in October 2013, it exited a profitable trade which it had not reentered, missing a good trend ride. Therefore it seems that some effective reentry technique won't hurt the performance. On a closing note, it would be intriguing to see if the K parameter could be made self-adjusting to the market's mood – as opposed to an arbitrary constant which it is now.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using TASCIndicators;

namespace WealthLab.Strategies
{
    public class QuotientTransformTASC201408 : WealthScript
    {
        private StrategyParameter paramPeriod;
        private StrategyParameter paramK1;
        private StrategyParameter paramK2;

        public QuotientTransformTASC201408()
        {
            paramPeriod = CreateParameter("Low-pass period",20,2,300,20);
            paramK1 = CreateParameter("K for entries",0.8,-1,1,1);
            paramK2 = CreateParameter("K for exits",0.4,-1,1,1);
        }

        protected override void Execute()
        {
            QuotientTransform qtEn = QuotientTransform.Series(Close,paramPeriod.ValueInt,paramK1.Value);
            QuotientTransform qtEx = QuotientTransform.Series(Close,paramPeriod.ValueInt,paramK2.Value);

            for(int bar = 2; bar < Bars.Count; bar++)
            {
                if (!IsLastPositionActive)
                {
                    if( CrossUnder( bar, qtEx, 0 ) )
                        SellAtMarket( bar+1, LastPosition );
                }
                else
                {
                    if( CrossOver( bar, qtEn, 0 ) )
                        if( BuyAtMarket( bar+1 ) != null )
                            LastPosition.Priority = -Close[bar];
                }
            }

            HideVolume();
            LineStyle ls = LineStyle.Solid;
            ChartPane pQT = CreatePane(40,true,true);
            PlotSeries(pQT,qtEx,Color.DarkCyan,ls,1);
            PlotSeries(pQT,qtEn,Color.FromArgb(255,255,140,0),ls,2);

            RSI rsi = RSI.Series(Close,14);
            DataSeries qrsi = (rsi - 50) / 50;
            ChartPane rp = CreatePane(40,false,true);
            qrsi.Description = "RSI Constrained";
            DataSeries qr = QuotientTransform.Series( qrsi, paramPeriod.ValueInt, paramK1.Value);
            PlotSeries(rp,qr,Color.Plum,ls,2);
        }
    }
}
```