

Matriks IQ İndikatörler

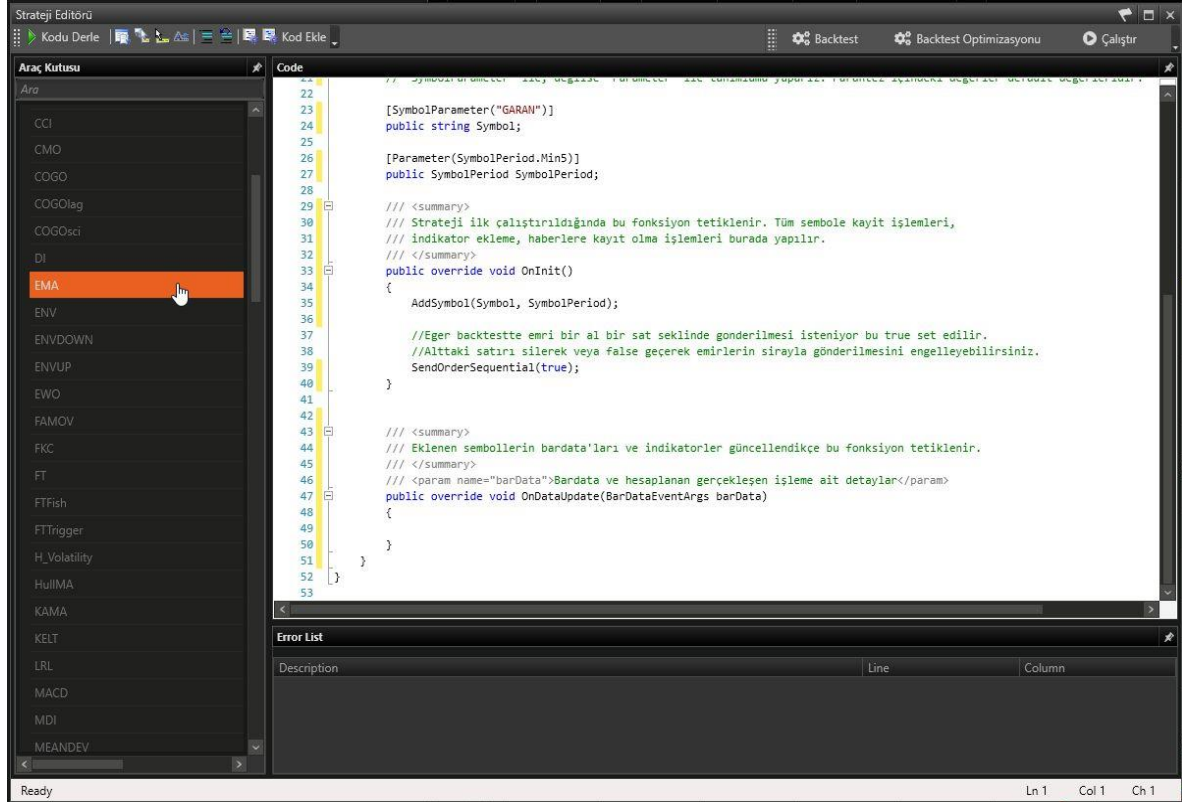
İçerik

Matriks IQ İndikatörler	1
ACCBandsIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal).....	4
AccumulationDistributionIndicator(String, SymbolPeriod, OHLCType).....	6
AccumulationDistributionOscillatorIndicator(String, SymbolPeriod, OHLCType, Int32, Int32)/ (IIndicator, Int32, Int32).....	7
ADXIndicator(String, SymbolPeriod, OHLCType, Int32).....	9
ATRIndicator(String, SymbolPeriod, OHLCType, Int32).....	10
BearPowerIndicator(String, SymbolPeriod, OHLCType, Int32, MovMethod).....	11
BollingerIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal, MovMethod).....	13
BullPowerIndicator(String, SymbolPeriod, OHLCType, Int32, MovMethod).....	15
CCIIndicator(String, SymbolPeriod, OHLCType, Int32).....	16
CenterOfGravityOscillatorIndicator(String, SymbolPeriod, OHLCType, Int32).....	18
CMOIndicator(String, SymbolPeriod, OHLCType, Int32).....	19
DIIndicator(String, SymbolPeriod, OHLCType, Int32).....	21
EMAIndicator(String, SymbolPeriod, OHLCType, Int32).....	23
EnvelopeIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal, MovMethod).....	25
EWOIndicator(String, SymbolPeriod, OHLCType, Int32, Int32).....	27
FAMOVIndicator(String, SymbolPeriod, OHLCType, Int32, Int32).....	29
FTIndicator(String, SymbolPeriod, OHLCType, Int32).....	31
FKCIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Decimal, Decimal).....	33
HullMAIndicator(String, SymbolPeriod, OHLCType, Int32).....	34
HVolatilityIndicator(String, SymbolPeriod, OHLCType, Int32).....	36
KAMAIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Int32).....	37
KELTIndicator(String, SymbolPeriod, OHLCType, Int32, Int32).....	39
LRLIndicator(String, SymbolPeriod, OHLCType, Int32).....	41
LRSIndicator(String, SymbolPeriod, OHLCType, Int32).....	43
MACDIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Int32).....	45
MDIIndicator(String, SymbolPeriod, OHLCType, Int32).....	47
MeanDevIndicator(String, SymbolPeriod, OHLCType, Int32).....	47
MOMIndicator(String, SymbolPeriod, OHLCType, Int32).....	48
MOSTIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal, MovMethod).....	50

MOVIndicator(String, SymbolPeriod, OHLCType, Int32, MovMethod).....	52
MSLIndicator(String, SymbolPeriod, OHLCType, Int32)	55
PDIndicator(String, SymbolPeriod, OHLCType, Int32).....	56
SMAIndicator(String, SymbolPeriod, OHLCType, Int32).....	59
STDEVIndicator(String, SymbolPeriod, OHLCType, Int32)	61
StochasticCCIIndicator(String, SymbolPeriod, OHLCType, Int32)	62
StochasticFastIndicator(String, SymbolPeriod, OHLCType, Int32, Int32).....	63
StochasticSlowIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Int32).....	65
TMAIndicator(String, SymbolPeriod, OHLCType, Int32).....	67
TSFIndicator(String, SymbolPeriod, OHLCType, Int32).....	69
VMAIndicator(String, SymbolPeriod, OHLCType, Int32)	71
VolumIndicator(String, SymbolPeriod)	73
VolumeTLIndicator(String, SymbolPeriod).....	75
WildersIndicator(String, SymbolPeriod, OHLCType, Int32)	76
WMAIndicator(String, SymbolPeriod, OHLCType, Int32)	78
YZVLTIndicator(String, SymbolPeriod, OHLCType, Int32).....	80
ZerolagIndicator(String, SymbolPeriod, OHLCType, Int32)	81

Platforma yeni indikatörlerin eklenmesi devam eden bir süreçtir. Bu dokümanda bazı indikatörler yer almaktadır.

MatriksIQ AlgoTrader'da birçok indikatör hazır şekilde sol taraftaki araç menüsünde bulunmaktadır.



Eklenmek istenen indikatör üzerine çift tıkladığında, otomatik olarak gerekli yerlerde o indikatörle ilgili tanımlar oluşturulmakta, indikatör stratejide kullanıma hazır hale gelmektedir.

Örneğin, EMA ögesine çift tıkladığında ilk kod bloğu içerisine `ema` isimli bir EMA ögesi ve `OnInit()` kod bloğu içerisine `ema = EMAIndicator(Symbol, SymbolPeriod, OHLCType.Close, 22);` satırı ile bu ögeye ait indikatör, varsayılan değerleriyle tanımlanmaktadır. Bu tanımlamalardan sonra `OnDataUpdate` kod bloğu içerisinde bu indikatör istenilen strateji uygulanarak kullanılabilir.

Her indikatörün genel çalışma prensibi ve stratejinin nasıl oluşturulabileceği dokümanın devamında anlatılmaktadır. MatriksIQ strateji editöründe [IntelliSense](#) de bulunduğundan, mevcut tanımlanmış indikatör objesinin ne metodlar alabileceğini CTRL+space'e basarak görebiliriz.

Örneğin yukarıda tanımlanmış `ema` objesi, `ema.CurrentIndex`, `ema.CurrentValue`, `ema.LastBarIndex`, `ema.Period`, `ema.Value` gibi birçok metod alabilmektedir.

- ** `ema.CurrentValue`: EMA indikatörünün o andaki değerini döndürür.
- ** `ema.CurrentIndex`: EMA indikatörünün istenilen andaki indeksini döndürür.
- ** `ema.LastBarIndex`: EMA indikatörünün istenilen andaki indeksini döndürür.
- ** `ema.Period`: EMA indikatörünün periodunu döndürür.

ACCBandsIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal)

(Acceleration bands) Hızlanma bantları göstergesinin değerini hesaplamak için kullanılır. Bollinger bantları gibi (genelde son 20 bar kullanır) basit averaj etrafında üst ve alt bant çizer. Enstrüman fiyatı bantların üstüne kırıldığında, alım yapılarak yukarı doğru hızlanması, altına kırıldığında da satış yapılarak aşağı doğru hızlanması beklenmektedir. Enstrüman fiyatı, tekrar bantların içine geldiğinde pozisyondan çıkılır.



Örnek:

```
if (CrossAbove(accBands.Lower, barData.BarData.Close))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
    Debug("Alış emri verildi.");
}
if (CrossBelow(accBands.Upper, barData.BarData.Close))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi.");
}
```

****accBands.Lower: Acceleration bands indikatörünün o andaki alt bant değeri**
****accBands.Upper: Acceleration bands indikatörünün o andaki üst bantı değeri**

AccumulationDistributionIndicator(String, SymbolPeriod, OHLCType)

Accumulation/distribution hacim ve fiyatı kümülatif olarak kullanarak, enstrümanın toplamda satım ya da alım altında olduğunu belirlemeye çalışan bir indikatördür. Eğer hisse fiyatı düşerken ADI yükseliyorsa, bu hissenin aslında toplanma durumunda olduğunu gösterdiğinden fiyatta da bir süre sonra yukarı hareket olacağı beklenmektedir.



Bu indikatör tek başına teknik bir strateji belirlemede yeterli değildir. O nedenle diğer indikatörlerle birlikte kullanılır.

AccumulationDistributionOscillatorIndicator(String, SymbolPeriod, OHLCType, Int32, Int32)/ (IIndicator, Int32, Int32)

Chaikin Accumulation/Distribution Oscillator göstergesinin değerini hesaplamak için kullanılır. Fiyat hareketleri ve işlem hacmini birlikte analiz ederek yön hakkında fikir vermeyi hedefler.




A/D Oscillator bir volume göstergesidir. Yürütülen mantık, fiyat hareketleri ile hacmin uyumuyla alakalıdır. Hacimdeki artışla fiyatlardaki yükselmenin ya da hacimdeki düşüşle fiyatlardaki düşüşün uyumlu hareket edeceği, aksi durumlarda ise trend değişikliklerinin meydana geldiği varsayılmaktadır. Göstergenin yorumlanmasında en yaygın yol, göstergenin yükselmesinin senette alım yapıldığı, düşmesinin ise senetten çıkıldığı şeklinde yorumlanmasıdır. Aynı şekilde, fiyatlar yükselirken bunun hacimle de desteklenmesi gerektiği varsayımından hareketle, Chaikin Oscillator göstergesinin yükselen fiyatlara aynı şekilde yükselerek cevap verememesi yükseliş trendinin sonlanabileceğine, düşen fiyatlara aynı şekilde düşerek cevap verememesi ise düşüş trendinin sonlanabileceğine işaret etmektedir.

Örnek:

```
if (temp != null)
{
    if (temp < accDistOscillator.CurrentValue)
    {

```



```
        SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
        Debug("Alış emri verildi.");
    }
    if (temp>accDistOscillator.CurrentValue)
    {
        SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
        Debug("Satış emri verildi.");
    }
    temp = accDistOscillator.CurrentValue;
}
else
{
    temp = accDistOscillator.CurrentValue;
}
```

****accDistOscillator.CurrentValue: Accumulation/Distribution Oscillator indikatörünün o andaki değeri**

ADXIndicator(String, SymbolPeriod, OHLCType, Int32)

Average Directional Movement Index göstergesinin değerini hesaplamak için kullanılır. J.W.Wilder tarafından geliştirilen Directional Movement, fiyatların hangi yönde hareket etme eğiliminde olduğunu araştırır. Trend belirleme ve yanlış sinyallerin filtre edilebilmesi açısından oldukça önemli bilgiler sunan Directional Movement, ADX, ADXR, DIS, DX ve DI+DI gibi birçok göstergenin de çıkış noktasıdır.



Hesaplanması kolay ama oldukça uzun olan Average Directional Movement Index (ADX) için öncelikle DM değerinin hesaplanması gerekmektedir. Bu değer, hareketin yönü yukarı ise pozitif aşağı ise negatif olacaktır. +DM ve -DM değerlerini bulduktan sonra ise bunların toplamlarını yine bunların farklarına bölerek DX değerine ulaşırız. Oldukça hızlı hareket eden DX göstergesinin 14 günlük hareketli ortalamasının alınarak yumuşatılmasıyla da ADX eğrisini elde etmiş oluruz. Average Directional Movement Index (ADX), 0 ile 100 arasında dolaşan bir göstergedir. Diğer birçok göstergeden farklı olarak ADX, alım satım sinyalleri üretmekten çok, bir trendin var olup olmadığı ve gücüyle ilgili bilgiler verir. ADX'in sıfıra yakın değerleri trendin olmadığına ve kararsız bir piyasayı göstermektedir. Gösterge değerinin artmaya başlaması ise fiyat hareketinin olduğu yönde bir trendin varlığına işaret etmektedir.

Bu indikatör tek başına teknik bir strateji belirlemede yeterli değildir. O nedenle diğer indikatörlerle birlikte kullanılır.

ATRIndicator(String, SymbolPeriod, OHLCType, Int32)

Average True Range göstergesinin deęerini hesaplamak için kullanılır. Fiyat hareketliliğinin artış / azalışı konusunda fikir verir.



J. Welles Wilder tarafından geliştirilen Average True Range göstergesi hisse senetlerinin hareketliliği üzerine kurulmuştur. Bu yüzden vereceği bilgi de senedin hareket yönünden bağımsız olarak, fiyatların ne kadar deęişkenlik göstermekte olduğudur.

Average True Range göstergesindeki yüksek deęerler fiyat hareketliliğindeki artışa işaret etmektedir. Bu ise trend dönüşlerinin zamanlamasında kullanılmaktadır. Nispeten düşük deęerler ise fiyatlardaki hareketliliğin azalmasıdır ve piyasadaki bir sıkışmayı göstermektedir.

Bu indikatör tek başına teknik bir strateji belirlemede yeterli deęildir. O nedenle dięer indikatörlerle birlikte kullanılır.

BearPowerIndicator(String, SymbolPeriod, OHLCType, Int32, MovMethod)

Bears Power bir aksi yönde trend indikatörüdür. Fiyatın yönünü belirlemenize ve o trendin dönüş yapacağı noktayı bulmanıza yarar.



Bears indikatöründe çubuklar eksi tarafta iken yükselme görülmeye başladığında alış sinyali verir. Satış sinyali ise bunun tam tersidir.

Örnek:

```
decimal temp;

if ((temp != null))
{
    if(temp>barData.BarData.Close && bearPower.CurrentValue>0)
    {
        SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
        Debug("Alış emri verildi.");
    }
    if (temp<barData.BarData.Close && bearPower.CurrentValue<0)
    {
        SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
        Debug("Satış emri verildi.");
    }
    temp = barData.BarData.Close;
}
else
{
    temp = barData.BarData.Close;
}
```

** bearPower.CurrentValue: Bear Power indikatörünün o andaki değeri

BollingerIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal, MovMethod)

Buradaki bantlar fiyatları saran kılıf gibidir. Daralmaya başlayan bant, yakında bant dışına doğru bir hareket olasılığı demektir. Yönü ve süresi belli değildir. Orta – Alt ve Üst olmak üzere 3 çizgiden oluşur.



Hareketli ortalamaların belli bir standart sapması alınarak hesaplandıktan sonra hareketli ortalamadan aşağı ve yukarı yönlü kaydırılarak çizilen Bollinger bantları durgun piyasalarda daralıp, hareketli piyasalarda genişleyerek farklı yorumlar yüklenebilecek sinyaller üretirler.

Bollinger bantlarındaki iki değişkenden biri olan periyot için Bollinger kendi uygulamalarında 20 günlük periyodu önerse de fiyat hareketleri daha az olan senetlerde daha kısa periyotlar kullanılabilir iken, aşırı fiyat hareketleri olan senetlerde daha uzun periyotlar da kullanılabilir.


Aynı periyot seçimlerinde olduğu gibi standart sapma değerinin seçiminde de incelemekte olduğunuz senedin hareketliliği belirleyici rol oynamaktadır. Nasıl ki fiyat salınımları daha az olan senetlerde hareketli ortalamamızın periyodunu küçültüyorsak aynı mantıkla standart sapma değeri de küçültülebilir, tersine aşırı fiyat hareketlerine sahip bir senette de aynı hareketli ortalama periyodunu büyüttüğümüz gibi standart sapma değerini de büyütebiliriz.

Bollinger bantlarının kullanımında fiyatların bantlar arasında gidip geldiği, bandın bir kenarına gelen fiyatların bunu takiben diğer banda doğru hareketlendiği ve bandın dışına taşan fiyat hareketlerinde de yeniden bandın içine dönüleceği varsayılır.

Bollinger bantlarındaki bir diğer önemli özellikse daralmalardır. Daralmaya başlayan bir bandın anlamı çok yakında fiyatlarda sert bir hareket olacaktır. Hareketlerin yönü hakkında kesin bir bilgi içermese de büyük marjlarda bir fiyat değişikliği uyarısıdır.

Örnek:

```
if (CrossAbove(bollinger.BollingerDown, barData.BarData.Close))
```



```
{  
    SendMarketOrder(Symbol, BuyOrderQuantity, (OrderSide.Buy));  
    Debug("Alış emri verildi.");  
}  
if (CrossBelow(bollinger.Bollingerup, barData.BarData.Close))  
{  
    SendMarketOrder(Symbol, SellOrderQuantity, (OrderSide.Sell));  
    Debug("Satış emri verildi.");  
}
```

** bollinger.BollingerDown: Bollinger bands indikatörünün alt bandının o andaki değeri

BullPowerIndicator(String, SymbolPeriod, OHLCType, Int32, MovMethod)

Bulls Power bir aksi yönde trend indikatörüdür. Bulls Power fiyatlara bakmak ve piyasanın ardındaki gücü görmek için kullanışlı bir yoldur. Bir trendin yönünü belirlemenize ve dönüş noktasını bulmanıza yarar. Histogram giderek azalıyor ancak henüz olumlu bölgeden çıkmadıysa, o zaman satış pozisyonları açmak için iyi bir andır.



CCIIndicator(String, SymbolPeriod, OHLCType, Int32)

Commodity Channel Index göstergesinin değerini hesaplamak için kullanılır.



D. Lambert tarafından mal piyasaları için geliştirilen ancak yapısı itibarıyla hisse senedi piyasaları için de uygun bir gösterge olan CCI da fiyatların istatistiksel ortalamadan ne ölçüde saptığı bulunmaya çalışılmaktadır.

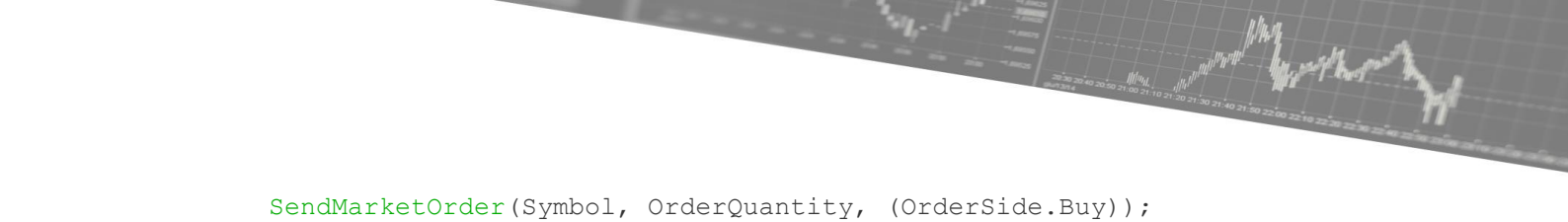
İstatistiksel ortalamadan sapma, trendin ve tercihlerin ne yönde değişmekte olduğunun anlaşılabilmesi açısından önemlidir. Daha çok kısa vadeli trend değişimlerini kovalayan ve yatay piyasalarda daha iyi sonuçlar veren CCI, +100 ve -100 de yer alan referans değerlerinin tanımladığı aşırı alım ve aşırı satım bölgelerinin kullanımına dayanan bir osilatördür.

Aşırı alım ve satım bölgelerini kullanan diğer göstergelerde olduğu gibi, bu bölgelerde dolaşan CCI değerleri bize trendin yakın bir gelecekte sonlanabileceği sinyallerini vermektedir. Aşırı alım bölgesi, fiyatların aşırı yükseldiğini ve her an satışların gelebileceğini söylerken bu bölgeden yapılacak alımların riskinin arttığını da anlatmaktadır. Aynı şekilde aşırı satım bölgesi de fiyatların aşırı düştüğünü ve burada alıma hazır olunması gerektiğini söylerken hala satmamış olanlar açısından satımın çok da mantıklı olmadığını anlatır.

CCI'yi yorumlamada kullanılan bir diğer metot ise fiyatlarla göstergenin uyumsuzluğudur. Fiyatlar bir önceki zirveyi geçerek yeni zirveler yaparken CCI yeni zirvesiyle buna eşlik edemiyor hatta bir önceki zirvenin altında kalıyorsa bu göstergenin yükselmeyi desteklemediğine bir işarettir ve fiyatlarda bir düzeltme beklenmelidir. Yeni dipler yapan fiyatlara yeni diplerle eşlik etmeyen bir CCI ise düşüşü desteklemiyor ve yukarı yönlü bir düzeltme sinyali veriyor demektir.

Örnek:

```
if (CrossAbove(cci, DownLevel))  
{
```

```
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
    Debug("Alış emri verildi.");
}
if (CrossBelow(cci, UpLevel))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi.");
}
```

**cci: Commodity Channel Index göstergesinin o andaki değeri
**DownLevel: Commodity Channel Index göstergesinin alt seviyesi
**UpLevel: Commodity Channel Index göstergesinin üst seviyesi

CenterOfGravityOscillatorIndicator(String, SymbolPeriod, OHLCType, Int32)

Destek ve direnç seviyeleri belirlemeye yardımcı olur. Birçok iyi bilinen göstergelerle karşılaştırıldığında yeni olmakla beraber gelecekteki fiyat salınım analizleri yapmak için uygun bir yöntem olarak çalışmaktadır, popülaritesi artmaktadır.



Bir osilatör olarak, başlıca iki faydası vardır: fiyata tepki olarak düşük gecikme ve net dönüm noktaları.

CMOIndicator(String, SymbolPeriod, OHLCType, Int32)

Chande's Momentum Oscillator göstergesinin değerini hesaplamak için kullanılır. Fiyatların yukarı / aşağı yönü konusunda sinyal verir.



Tushar Cande tarafından geliştirilen bir Momentum göstergesi olan CMO, belli bir zaman dilimi içerisinde fiyatların ne yönde ve ne şiddette hareket ettiğini gösterir.

Momentum'daki 100 referans değerinin etrafında salınan eğrinin yerini Chande's Momentum Oscillator'da 0'ın etrafında salınan bir eğri almıştır. CMO değerinin "0"ın üzerinde olduğu değerler fiyatların yukarı yönlü arzusunun anlatırken, altındaki her değer fiyatlardaki düşüş eğilimine işaret eder.

CMO da kullanılan diğer iki referans çizgisi ise +50 ve -50 değerleridir. +50 seviyesinin üzerindeki değerler aşırı alımı ve fiyatların yakın bir zamanda düşüş gösterebileceğine işaret ederken, -50 seviyesinin altındaki değerler ise aşırı satımı ve fiyatların yakın zamanda yükselme potansiyelini göstermektedir.

CMO yorumlanmasında kullanılan bir diğer unsur da uyumsuzluklardır. Uyum, gevşeyen fiyatlar ve alçalan trend ile beraber oluşmakta olan yeni diplerin ve tepelerin bir öncekinden daha düşük seviyelerde oluşması, yükselen fiyatlar ve yükselen trend ile beraber de yeni oluşan tepelerin ve diplerin bir öncekinden yukarıda olması demektir. Oluşumun bu şekilde gelişmediği durumlar uyumsuzluk olarak adlandırılır ve yakın bir zamanda trendin gücünü kaybederek ters yönde bir hareket yapacağı kabul edilir.

Örnek:

```
if (CrossAbove(cmo, DownLevel))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
    Debug("Alış emri verildi.");
}
if (CrossBelow(cmo, UpLevel))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi.");
}
```

****cmo: Chande's Momentum Oscillator göstergesinin o andaki değeri**
****DownLevel: Chande's Momentum Oscillator göstergesinin alt seviyesi**
****UpLevel: Chande's Momentum Oscillator göstergesinin üst seviyesi**

DIIndicator(String, SymbolPeriod, OHLCType, Int32)

Directional Indicator göstergesinde pozitif ve negatif olmak üzere 2 ayrı çizgi vardır.




J.W.Wilder tarafından geliştirilen Directional Movement fiyatların hangi yönde hareket etme eğiliminde olduğunu araştırır. Trend belirleme ve yanlış sinyallerin filtre edilebilmesi açısından oldukça önemli bilgiler sunan Directional Movement, ADX, ADXR, DIS, DX ve DI+DI- gibi birçok göstergenin de çıkış noktasıdır.

Hesaplanması kolay ama oldukça uzun olan Directional Movement için öncelikle DM değerinin hesaplanması gerekmektedir. Bu değer, bir önceki güne göre hareketin yönü yukarı ise pozitif aşağı ise negatif olacaktır. +DM ve -DM değerlerini bulduktan sonra ise +DM değerinden DI+, -DM değerinden de DI- türetilir.

DI+ ve DI- grafik üzerinde beraber çizilerek gösterilirler. Fiyatın yükseldiği dönemlerde pozitif yönlü hareketin bir türevi olan DI+ yükselirken negatif yönlü hareketin bir türevi olan DI- düşecektir. Farkın DI+ lehine açılması yukarı yönlü bir trendin varlığına işaret etmektedir. Aynı şekilde fiyatların düştüğü dönemlerde bu sefer DI- artacak ve DI+ düşecektir. Farkın DI- lehine açılması ise aşağı yönlü bir trendin varlığını göstermektedir.

Yorumlama açısından genel kullanım DI+ göstergesinin DI-'yi yukarı doğru keserek üzerine çıkmasıyla "al" sinyalinin üretildiği, DI- eğrisinin DI+ eğrisini yukarı doğru keserek üzerine çıkmasıyla ise "sat" sinyalinin üretildiği şeklindedir. Ancak trendin varlığında oldukça etkili alım satım noktaları veren DI+DI- trendin olmadığı dönemlerde çok sık kesişerek hatalı sinyaller de üretebilmektedir.



Al ve sat noktalarına biraz daha temkinli yaklaşan ve hatalı sinyalleri filtre etmeyi amaçlayan bir diğel yöntemde ise DI+'nın DI-'yi yukarı keserek "al" verdiđi noktada hemen alım yapılmamalıdır.

Kesişim gününün görülen en yüksek fiyatının bir sonraki işlem gününde geçilmesi beklenmelidir. Aynı şekilde DI-'nin DI+'yı yukarı keserek "sat" verdiđi noktada da satım yapılmamalı ve bir sonraki iş gününde fiyatın kesişim gününde gördüğü en düşük deđerin altına inmesi beklenmelidir.

EMAIndicator(String, SymbolPeriod, OHLCType, Int32)

Exponential Moving Average (EMA) göstergesinin deęerini hesaplamak için kullanılır.



Üssel Hareketli Ortalama (EMA), hareketli ortalamayla çok benzerdir (ve bir türüdür). Hareketli ortalamada basit ortalama kullanılırken Üssel Hareketli Ortalamada exponential ortalama kullanılır.

Örnek:

```
decimal prevValue;

if (prevValue != null)
{
    if (prevValue < ema.CurrentValue)
    {
        SendMarketOrder(Symbol, BuyOrderQuantity, (OrderSide.Buy));
        Debug("Alış emri verildi.");
    }
    if (prevValue > ema.CurrentValue)
    {
        SendMarketOrder(Symbol, SellOrderQuantity, (OrderSide.Sell));
        Debug("Satış emri verildi.");
    }
    prevValue = ema.CurrentValue;
}
else
{
    prevValue = ema.CurrentValue;
}
```

****ema.CurrentValue: Exponential Moving Average (EMA) göstergesinin o andaki değeri**

**** prevValue: Exponential Moving Average (EMA) göstergesinin bir önceki değeri**

EnvelopeIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal, MovMethod)

Bollinger bandına benzer ama daha basitidir. Hareketli ortalamanın belli bir miktar üstü ve altından geçen 2 ad çizgisi vardır.




Envelopes'in iki değişkeninden biri olan hareketli ortalama periyoduna karar verirken aynı Bollinger da olduğu gibi fiyat hareketleri daha az olan senetlerde daha kısa periyotlar kullanılabilir iken, aşırı fiyat hareketleri olan senetlerde daha uzun periyotlar da kullanılabilir.

Bantların aşağı ve yukarı yönde ne ölçüde kaydırılacağını gösteren kaydırma oranının belirlenmesinde de fiyat hareketliliğine bakılması ve fiyat hareketleri daha az olan senetlerde daha küçük kaydırma oranları belirlenirken, aşırı fiyat hareketleri olan senetlerde daha büyük kaydırma oranlarının belirlenmesi gerekmektedir.

Envelopes'in yorumlanmasında fiyat salınımlarının bant içinde kaldığı ve bir banda ulaşan fiyat hareketlerinin buradan dönerek diğer banda doğru hareket edeceği varsayılır. Bu şekliyle bantların destek direnç gibi algılandığı söylenebilir. Alt banda ulaşılması "al", üst banda ulaşılması ise "sat" sinyali üretmektedir.

Örnek:

```
var barDataModel = GetBarData();  
  
if (CrossAbove(barDataModel, env.Down, OHLCType.Close))
```

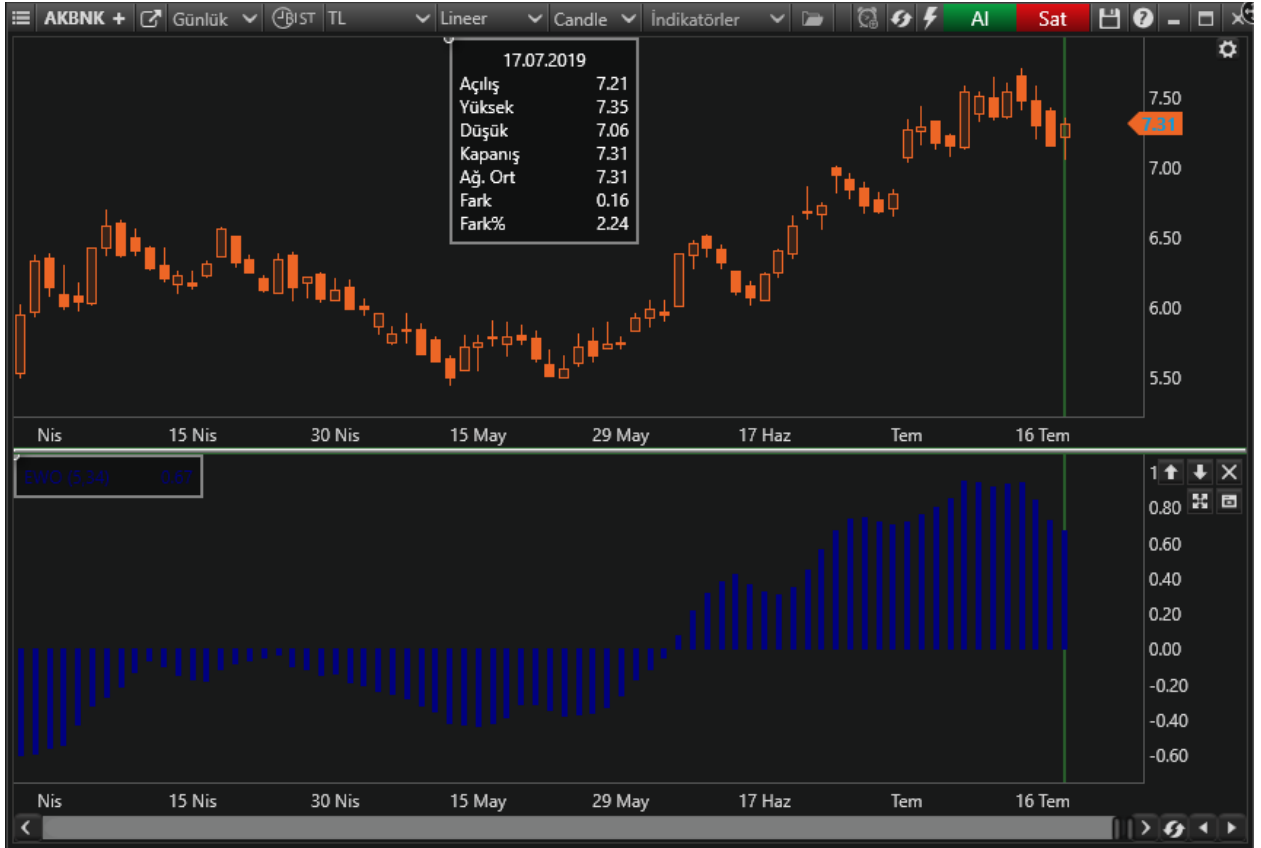


```
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
    Debug("Alış emri verildi.");
    Debug("Close:" + barData.BarData.Close);
    Debug("Env.Down:" + env.Down.CurrentValue);
    Debug("Env.Up:" + env.Up.CurrentValue);
}
if (CrossBelow(barDataModel, env.Up, OHLCType.Close))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi.");
    Debug("Close:" + barData.BarData.Close);
    Debug("Env.Down:" + env.Down);
    Debug("Env.Up:" + env.Up);
}
```

** env.Down.CurrentValue: Envelope göstergesinin Down bandının değeri
** env.Up.CurrentValue: Envelope göstergesinin Up bandının değeri

EWIndicator(String, SymbolPeriod, OHLCType, Int32, Int32)

Elliott wave oscillator göstergesinin değerini hesaplamak için kullanılır.



Elliott Wave Oscillatör, 5 ve 34 günlük basit ortalamalar arasındaki farktır. Gösterim şekli olarak 'Histogram' seçilmiştir. Fiyat hareketlerinin dalgalar şeklinde ilerlediğini varsayar. Yeni zirve, bir önceki zirvenin üzerinde oldukça, yükselişin devam edeceğini, altında kaldığında ise, trendin düşüşe döneceğini varsayar.

Örnek:

```
if (CrossAbove(ewo, 0))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
    Debug("Alış emri verildi.");
}
if (CrossBelow(ewo, 0))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi.");
}
```

**ewo: Elliott wave oscillator göstergesinin o andaki değeri

FAMOVIndicator(String, SymbolPeriod, OHLCType, Int32, Int32)

Fraktal Uyarlamalı Hareketli Ortalama (FRAMA), John Ehlers tarafından geliştirilen akıllı, uyarlanabilir bir hareketli ortalamadır. Fiyat değişimlerinin önemini dikkate alır ve fiyat dalgalanırken, düz kalırken belirgin değişimlerde fiyatı yakından takip eder. FRAMA, piyasaların fraktal olduğu gerçeğinden yararlanarak bu fraktal geometriye dayalı olarak geriye doğru inceleme süresini dinamik olarak ayarlar. Hesaplama çok ayrıntılı ve karmaşıktır. FRAMA genellikle diğer sinyaller ve analiz teknikleri ile birlikte kullanılır.



Örnek:

```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, famov, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, (OrderSide.Buy));
    Debug("Alış emri gönderildi.");
    Debug("Close = " + barData.BarData.Close);
    Debug("Famov = " + famov.CurrentValue);
}
if (CrossBelow(barDataModel, famov, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, (OrderSide.Sell));
    Debug("Satış emri gönderildi.");
    Debug("Close = " + barData.BarData.Close);
    Debug("Famov = " + famov.CurrentValue);
}
```

****famov.CurrentValue: Fraktal Uyarlamalı Hareketli Ortalama göstergesinin o andaki değeri**
****OHLCType.Close: Barların kapanış değerlerine göre işlem yapılmasını sağlar**

FTIndicator(String, SymbolPeriod, OHLCType, Int32)

Fisher Transform(FT) indikatörünün 2 ayrı çizgisi vardır.



Ehler tarafından tasarlanmış bir indikatördür. Çizmiş olduğu 2 tane çizginin kesişme yerine göre sinyal verir. Yeşil çizginin sarı çizgiyi yukarı doğru kırmaması al sinyali üretirken aşağı doğru kırmaması sat sinyali üretir.

Örnek:

```
if (CrossAbove(ft.Fish, ft.Trigger))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, (OrderSide.Buy));
    Debug("Alış emri verildi."); Debug("FtFish:" + ft.Fish.CurrentValue);
    Debug("FtTrigger" + ft.Trigger.CurrentValue);
}
if (CrossBelow(ft.Fish, ft.Trigger))
{
    SendMarketOrder(Symbol, SellOrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi."); Debug("FtFish:" + ft.Fish.CurrentValue);
    Debug("FtTrigger" + ft.Trigger.CurrentValue);
}
```

** ft.Fish.CurrentValue: Fisher Transform göstergesinin Fish bandının değeri

** ft.Trigger.CurrentValue: Fisher Transform göstergesinin Trigger bandının değeri

FKIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Decimal, Decimal)

Fraktal Keltner kanalları uyarlanabilir bir fraktal hareketli ortalamaya odaklanır. Yeşil kanala yukarıdan giren bir çubuk kısa bir eğilim gösterirken mavi kanala alttan gelen sinyallere uzun bir eğilim izler. Önceki kanalının tamamen dışında kalan bir bar, trendin sonunu gösterir.



HullMAIndicator(String, SymbolPeriod, OHLCType, Int32)

Hull Moving Average (HMA) göstergesinin değerini hesaplamak için kullanılır.




Hull Hareketli Ortalaması (HMA), Alan Hull tarafından gecikmeyi azaltmak, yanıt vermeyi artırmak ve aynı zamanda gürültüyü ortadan kaldırmak amacıyla geliştirilmiştir. Hesaplaması ayrıntılıdır ve Ağırlıklı Ortalama(WMA) hesabını kullanır. Gelecekteki piyasa trendini belirlemek için, hızlı hareket eden bir hareketli ortalama üreten eski fiyatlar üzerinden son fiyatları vurgular. Giriş ve çıkış sinyalleri için de kullanılabilir. Gösterge çoğunlukla salınım tacirleri ve uzun vadeli işlem yapanlar tarafından diğer sinyaller ve analiz teknikleriyle birlikte kullanılır.

Örnek:

```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, hullMA, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("hullMA:" + hullMA.CurrentValue);
}
if (CrossBelow(barDataModel, hullMA, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
}
```



```
Debug("Satış Emri Gönderildi");
Debug("Close:" + barData.BarData.Close);
Debug("hullMA:" + hullMA.CurrentValue);
}
```

** hullMA.CurrentValue: Hull Moving Average göstergesinin o andaki değeri

HVolatilityIndicator(String, SymbolPeriod, OHLCType, Int32)

Volatility göstergesinin deęerini hesaplamak için kullanılır.



Historical (Tarihi) Volatility (Hareketlilik), son bir yılda gerçekleşen günlük bazdaki volatilitayı ölçer. Normal volatilitenin aksine geçmişteki gerçekleşmeye bakmaktadır.

Bu indikatör tek başına teknik bir strateji belirlemede yeterli değildir. O nedenle diğer indikatörlerle birlikte kullanılır.

KAMAIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Int32)

Kaufman's Adaptive Moving Average (KAMA), Perry Kaufman tarafından geliştirilen akıllı bir hareketli ortalamadır. Güçlü bir trend takip göstergesi olup Üssel Hareketli Ortalama (EMA)'ya dayanır, trend ve volatiliteye karşı duyarlıdır.




Geleneksel hareketli ortalamaların en zayıf noktalarından biri, alım satım sinyalleri için kullanıldığında birçok yanlış sinyal üretme eğiliminde olmalarıdır. KAMA göstergesi bu eğilimi azaltmayı hedefler. Daha az yanlış sinyal üretir. Kısa vadeli, önemsiz fiyat hareketlerine cevap vermez.

Gürültü düşük olduğunda fiyatı yakından takip eder ve fiyat dalgalandığında gürültüyü düzleştirir. Tüm hareketli ortalamalar gibi, KAMA da trendi görselleştirmek için kullanılabilir. Fiyat kesişmesi, bir yön değişimini belirtir. Fiyat, dinamik destek ve direnç noktaları olarak görülebilecek şekilde KAMA'dan sıçramalar yapabilir. Genellikle diğer sinyaller ve analiz teknikleri ile birlikte kullanılır.

Örnek:

```
var barDataModel = GetBarData();  
  
if (CrossAbove(barDataModel, kama, OHLCType.Close))  
{  
    SendMarketOrder(Symbol, BuyOrderQuantity, (OrderSide.Buy));  
}
```



```
    Debug("Alış emri verildi."); Debug("KAMA:" + kama.CurrentValue);
    Debug("Bardata.Close:" + barData.BarData.Close);
}
if (CrossBelow(barDataModel, kama, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi."); Debug("KAMA:" + kama.CurrentValue);
    Debug("Bardata.Close:" + barData.BarData.Close);
}
```

** kama.CurrentValue: Kaufman's Adaptive Moving Average göstergesinin değeri

KELTIndicator(String, SymbolPeriod, OHLCType, Int32, Int32)

Keltner Kanalları göstergesinin değerini hesaplamak için kullanılır.




Keltner Kanalları göstergesi, Bollinger Bantları ve Hareketli Ortalama Zarflarına benzer bantlı bir göstergedir. Bunlar Orta Çizginin üzerinde ve altındaki birer zarftan oluşur. Orta Çizgi, kullanıcı tanımlı bir zaman aralığında hesaplanan fiyatın bir hareketli ortalamasıdır. Genellikle basit hareketli ortalama veya üstel hareketli ortalama kullanılır. Üst ve Alt Zarflar (kullanıcı tanımlı), Orta Çizgiden bir mesafede olarak ayarlanır. Bu günlük yüksek / düşük aralığının katları veya daha yaygın olarak Ortalama Gerçek Aralık'ın bir katı olabilir.

Örnek:

```
if (barData.BarData.Close < kelt.KeltDown.CurrentValue)
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
    Debug("Alış Emri Gönderildi");
}

if (barData.BarData.Close > kelt.KeltUp.CurrentValue)
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
    Debug("Satış Emri Gönderildi");
}
```



****kelt.KeltDown.CurrentValue:** Keltner Kanalları göstergesinin alt çizgisinin o andaki değeri

****kelt.KeltUp.CurrentValue:** Keltner Kanalları göstergesinin üst çizgisinin o andaki değeri

****kelt.CurrentValue:** Keltner Kanalları göstergesinin o andaki değeri

LRLIndicator(String, SymbolPeriod, OHLCType, Int32)

Linear Regression göstergesinin deęerini hesaplamak için kullanılır.



Doęrusal baęlanım olarak da tanımlanan bu gösterge, fiyatların Linear regression çizgisinin üzerinde kaldığı süreci AL sinyalini, altına düřtüęü zamanlarda ise SAT sinyalini işaret eder.

Örnek:

```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, lrl, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("lrl:" + lrl.CurrentValue);
}

if (CrossBelow(barDataModel, lrl, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("lrl:" + lrl.CurrentValue);
}
```

**lrl.CurrentValue: Linear Regression göstergesinin o andaki değeri

LRSIndicator(String, SymbolPeriod, OHLCType, Int32)

Linear Regression Slope göstergesinin deęerini hesaplamak için kullanılır.



Linear Regression'dan üretilen bu grafik, sıfırın üstüne çıktığında AL sinyali, sıfırın altına düştüğünde ise SAT sinyalini işaret eder.

Örnek:

```
if (CrossAbove(lrs, 0))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, (OrderSide.Buy));
    Debug("Alış emri verildi."); Debug("LRS:" + lrs.CurrentValue);
    Debug("Bardata.Close:" + barData.BarData.Close);
}
if (CrossBelow(lrs, 0))
{
    SendMarketOrder(Symbol, SellOrderQuantity, (OrderSide.Sell));
    Debug("Satış emri verildi.");
    Debug("LRS:" + lrs.CurrentValue);
    Debug("Bardata.Close:" + barData.BarData.Close);
}
```

** lrs.CurrentValue: LRS göstergesinin o andaki değeri

MACDIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Int32)

MACD göstergesinin değerini hesaplamak için kullanılır.



Genelde 26 ve 12 günlük olarak kullanılan iki üssel hareketli ortalamanın farkı olan MACD, kısa vadeli üstel ortalamanın uzun vadeliye göre olan pozisyonlarını değerlendirerek trendin yönü hakkında bilgi vermeye çalışır.

Macd'nin 0 olduğu seviyede 12 günlük üssel hareketli ortalama 26 günlük üssel ortalamaya eşittir. 12 günlüğün 26 günlüğün üzerine çıktığı durumlarda ise Macd pozitif değerler alacaktır. Tersine 12 günlüğün 26 günlüğün altına indiği durumlarda ise Macd negatif olacaktır.

Yorumlamalarda 12 günlük hareketli ortalamanın 26 günlük hareketli ortalamadan uzaklaştığı yani Macd'nin 0 değerinin altında ya da üzerinde olduğu durumlar aşırı alım ve aşırı satımlar olarak değerlendirilebilir. Ancak bu bölgelerde olmanın ürettikleri sinyaller referans değeri kullanan diğer göstergelere göre daha zayıftır. Fiyatların trend yönünde aşırı bir hareketi olduğunu söylese de dönüş zamanı hakkında kesin bilgiler içermez.

Macd eğrisinin 9 günlük hareketli ortalamasının alınarak buna göre olan pozisyonların değerlendirilmesidir. Bu yöntemde Macd eğrisi, 9 günlük hareketli ortalaması olan trigger eğrisini yukarı yönlü kestiğinde "al", aşağı yönlü kestiğinde ise "sat" sinyali verilmiş olur.

Macd'nin yorumlanmasında önem taşıyan bir diğer yöntemde uyumsuzlukların araştırılmasıdır. Uyumun, gevşeyen fiyatlar ve alçalan trend ile beraber oluşmakta olan yeni diplerin ve tepelerin bir öncekinden daha düşük seviyelerde oluşması, yükselen fiyatlar ve yükselen trend ile beraber de yeni oluşan tepelerinin ve diplerinin bir öncekinden yukarıda

olması demek olduğunu kabul edersek oluşumun bu şekilde gelişmediği durumlar uyumsuzluk olarak adlandırılır ve yakın bir zamanda trendin gücünü kaybederek ters yönde bir hareket yapacağı kabul edilir.

Örnek:

```
if (CrossAbove(macd, macd.MacdTrigger))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
}
if (CrossBelow(macd, macd.MacdTrigger))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
}
```

****macd.MacdTrigger: MACD göstergesinin Trigger bandının o andaki değeri**

MDIIndicator(String, SymbolPeriod, OHLCType, Int32)

McGinley Dynamic göstergesi (MDI), piyasayı mevcut hareketli ortalama göstergelerinden daha iyi izlemek için tasarlanmış bir hareketli ortalama türüdür.

Piyasa hızındaki değişimleri ayarlayarak hareketli ortalama çizgilerini iyileştiren teknik bir göstergedir. MDI çizgisi, volatilité yükselmesi olmadıkça fiyatlar ile birlikte hareket eder.

Örnek:

```
if (CrossAbove(mdi, barData.BarData.Close))
{
    SendMarketOrder(Symbol, OrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
}
if (CrossBelow(mdi, barData.BarData.Close))
{
    SendMarketOrder(Symbol, OrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
}
```

MeanDevIndicator(String, SymbolPeriod, OHLCType, Int32)

Mean Deviation göstergesi William Blau tarafından yazılmıştır. MeanDev göstergesi, fiyat değerini ve üssel hareketli ortalama (EMA) farkını kullanır.

Mean Deviation x dönem hareketli ortalama fiyat konumunu gösterir. MeanDev değeri, pozitif ise aşağı yönlü trend, hareketli ortalamanın altında ise negatif yönlü trend vardır.

MOMIndicator(String, SymbolPeriod, OHLCType, Int32)

Momentum göstergesinin değerini hesaplamak için kullanılır.



Momentum, belli bir zaman dilimi içinde fiyatların ne yönde ne miktarda ve ne şiddette hareket ettiğini anlatan bir göstergedir. Momentumun asıl hedefi, periyot kadar önceki kapanışa göre şu anki kapanışın nerede olduğunu bulmasıdır.

Kısa vadeli bir gösterge olarak kullanacağımız momentumda 12- 14 günlük periyotlar daha olumlu sonuçlar vermesine karşın kullanacağınız vade, ilgilenilen senede ve uygulanan stratejilere göre değişebilmektedir.

Hareketliliğin yüksek olduğu senelerde vade uzatılabilirken hareketlilik azaldıkça vade de kısaltılabilir.

Son günün kapanışının, x gün önceki kapanışı bölünmesinin 100 ile çarpımı olarak hesaplanan Momentum'da bu iki günün eşit olduğu durumlarda bulunacak olan 100 değeri referans değeri olarak kabul edilir.

Kullanım açısından diğer referans değerleri kullanan indikatörlerde olduğu gibi bu seviyenin üzerinde "aşırı alım" altında ise "aşırı satım" olarak kabul edilmesi mümkündür. Ayrıca Momentumla bu çizginin kesişme noktalarında Momentumun referans değerini yukarı kesmesi "al" aşağı kesmesi ise "sat" sinyali olarak kabul edilebilir.

Momentumun yorumlanmasında bir diğer unsur da uyumsuzluklardır. Uyum, gevşeyen fiyatlar ve alçalan trend ile beraber oluşmakta olan yeni diplerin ve tepelerin bir öncekinden daha düşük seviyelerde oluşması, yükselen fiyatlar ve yükselen trend ile beraber de yeni

oluşan tepelerinin ve diplerin bir öncekinden yukarıda olması demektir. Oluşumun bu şekilde gelişmediği durumlar uyumsuzluk olarak adlandırılır ve yakın bir zamanda trendin gücünü kaybederek ters yönde bir hareket yapacağı kabul edilir.

Örnek:

```
if (CrossAbove(mom, ema))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
}
else if (CrossBelow(mom, ema))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
}
```

MOSTIndicator(String, SymbolPeriod, OHLCType, Int32, Decimal, MovMethod)

MOST göstergesinin değerini hesaplamak için kullanılır.



Anıl Özekşi tarafından geliştirilen Moving Stoploss göstergesi, adından da anlaşılacağı üzere hareketli stoploss mantığı üzerine inşa edilmiş bir modeldir. Ancak bu göstergedeki hareketli stoploss, klasik stoploss tanımından farklı olarak, fiyatların düşmesi durumunda alınmış olan pozisyonun kapatılması eyleminin yanına fiyatların yükselmesi durumunda kapatılmış olan pozisyonun yeniden açılması eylemini de ekleyerek fiyat salınımlarına mükemmele uyan ve bu salınımları bir kılıf gibi örten bir yapıda sunulmaktadır.

Klasik anlayıştaki hareketli stoploss mantığından ayıran bir diğer özelliği ise, MOST göstergesinin direk fiyatlarla ilişkilendirilmeyerek al ve sat sinyallerini fiyatların üç günlük üstel ortalaması yardımıyla üretiyor olmasıdır. Fiyat hareketlerindeki aşırılıkların bir ölçüde filtrelendiği bu üç günlük üstel ortalamanın MOST eğrisini yukarı doğru keserek üzerine çıkması al sinyali, aşağı doğru keserek altına inmesi ise sat sinyali olarak kabul edilir. Bu ise, üç günlük üstel ortalamanın üstte olduğu tüm durumlar için alınmış pozisyonların korunacağı, MOST eğrisinin üstte olması durumunda ise satış yapılarak nakit pozisyonda kalınacağı anlamına gelecektir. Burada bir diğer önemli nokta ise MOST'un sadece fiyat hareketlerine odaklanmış bir gösterge olmasıdır. Yani çok kısa vadeli hareketli ortalamamızı yumuşatılmış fiyatlar, MOST'u ise bir hareketli ortalama gibi düşünecek olursak, bu göstergenin çalışma prensibinin hareketli ortalamalardaki gibi olduğu görülecektir. MOST, fiyatları temsil eden hareketli ortalamanın üzerine çıkarsa satım, altına inerse de alım yapılacaktır. Tek fark, MOST kendi ile aynı yöndeki hareketli ortalama hareketinde aradaki stoploss mesafesini koruyarak ortalamayı izlerken, hareketli ortalama oluşun ters yönlü bir harekete yataya girerek cevap vermektedir.

MOST eğrisindeki iki değişkenden ilki ilişki içinde olduğu hareketli ortalamanın periyodudur. Yani fiyatlar için kullanmış olduğunuz yumuşatma oranı ne ise MOST'un periyodu da o olacaktır. Burada üç günlük bir ortalama seçmemizin nedeni ise, daha kısa vadelerin bizi fiyatların aşırı hareketlerini filtre etme amacından uzaklaştırması, daha uzun vadelerde ise hareketli ortalamanın fiyatları taklit kabiliyetinin azalmasıdır.

MOST eğrisini oluşturan ikinci değişken ise, kullanılan hareketli ortalama ile sinyalizasyonunu sağlayan kaydırma oranı yani stoploss mesafesidir. Analizlerimizde varsayılan olarak %2 kabul ettiğimiz bu kaydırma oranı da diğer göstergelerde olduğu gibi vadeli tercihlere konu olabilse de yarım puanlık oynamalara bile duyarlı olduğu ve küçülmesinin hatalı sinyal, büyümesinin ise üretilecek sinyallerde gecikme riski taşıdığı unutulmamalıdır.

Örnek:

```
if (CrossAbove (most, most.ExMOV))
{
    SendMarketOrder (Symbol, OrderQuantity, (OrderSide.Buy));
    Debug ("Alış emri verildi.");
    Debug ("Most.ExMov:" + most.ExMOV.CurrentValue);
    Debug ("Most:" + most.CurrentValue);
}
if (CrossBelow (most, most.ExMOV))
{
    SendMarketOrder (Symbol, OrderQuantity, (OrderSide.Sell));
    Debug ("Satış emri verildi.");
    Debug ("Most.ExMov:" + most.ExMOV.CurrentValue);
    Debug ("Most:" + most.CurrentValue);
}
```

**most: Moving Stoploss göstergesinin o andaki değeri

**most.ExMov: Moving Stoploss göstergesinin ExMov bandının o andaki değeri

MOVIndicator(String, SymbolPeriod, OHLCType, Int32, MovMethod)

Hareketli ortalama göstergesinin değerini hesaplamak için kullanılır.



Hareketli Ortalama, herhangi bir hisse senedindeki fiyatların, belli bir zaman aralığındaki ortalamasıdır. Örneğin kapanış verisinin 22'lik basit hareketli ortalamasının hesaplanması şu şekildedir: Fiyat serisi üzerindeki her bar için, kendi dahil olmak üzere geriye doğru toplam 22 barın kapanışlarının toplanarak 22 ye bölünmesi suretiyle elde edilmiştir.

Bir hareketli ortalamının hesaplanabilmesi için öncelikle üç değişken üzerinde karar vermemiz gerekir. Bunlar veri tipi, periyot ve hesaplama şeklidir.

Veri tipi: Hareketli ortalama hesaplamasında kullanacağımız ilk değişken veri tipidir. Açılış, kapanış, en yüksek, en düşük, ağırlıklı ortalama ya da başka bir veri kullanarak hareketli ortalamayı hesaplamak mümkündür. Kullanım açısından en çok kullanılanı kapanışa göre hesaplamaktır.

İpucu: Veri olarak bir başka indikatörü kullanabilir ve o indikatörün hareketli ortalamasını da hesaplayabilirsiniz.

Periyot: Periyodun seçimi daha çok ne tür bir yatırım stratejisi ile ilgilendiği ile alakalıdır. Hareketli ortalamayı kısa vadeli alım satımlar için kullanmak isteyenlerin tercihleri çoğunlukla 5, 9 ya da 14 günlük ortalamalar olurken orta ve uzun vadeli yatırımcıların tercihleri 50, 100 günlük olur. Daha uzun vadeli bakanlar ise 200 ve üzeri hareketli ortalamalar kullanmaktadır.

İpucu: Bunlar kesin değerler değildir. Ayrıca, yakın vadeli bakanlar 5-60 dakikalık periyot kullanabilir. Uzun vadeli bakanlar da Haftalık periyot kullanabilirler.

Hesaplama yöntemi: Hareketli ortalama yöntemi oldukça çok sayıda seçenek vardır. Bunlar hesaplama yaptıkları barların verilerini farklı yöntemlerle hesaplamaya dahil ederler.

Yöntemler:

S: Simple (Basit)) (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir)
E: Exponential (Üstel) (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir)
W: Weighted (Ağırlıklı) (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir)
TRI: Triangular (Üçgensel) (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir)
VAR: Variable (Değişken) (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir)
ZL: Zero Lag (Yakınsayan Diyebiliriz) (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir)
WW: Welles Wilder (Hesaplayan kişinin adı ile anılmaktadır) (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir.)
TSF: Time Series Forecast (Aynı İsimde Bir İndikatör var. Talep üzerine MatriksIQ' ya eklenmiştir.)

Ancak son günlerin önemini atlayan bu yaklaşıma karşı ağırlıklı hareketli ortalama da ilk günlerin önemi azaltılarak ağırlık son günlere kaydırılır. Üssel hareketli ortalama da ise ağırlığın yine son günlere verilmesine karşın ilk günler ağırlıklı hareketli ortalama olduğu gibi ihmal edilmez.

Diğer yöntemler biraz daha az kullanılmaktadır. Üçgensel hareketli ortalama da ağırlık orta kısımlara verilir. Değişken hareketli ortalama da hesaplamanın içine bir de dalgalanma oranı dahil edilmiştir. Zero Lag fiyatlara oldukça yakınsayarak gider.

En basit hareketli ortalama yorumu fiyatların hareketli ortalama ile karşılaştırılmasıdır. Buna göre fiyatların hareketli ortalamanın altına inmesiyle "sat", üzerine çıkmasıyla ise "al" sinyali üretilmiş olur. Burada amaç en dip ve en tepeleri yakalamaktan çok trend dönüşlerini yakalamak ve trend süresince trendin gerektirdiği pozisyonda kalmaya çalışmaktır.

Hareketli ortalama alım ve satım sinyallerini yorumlarken kullandığımız bir diğer yöntem ise farklı periyotta iki hareketli ortalama birden kullanarak bunların hareketlerini hem birbirleriyle hem de fiyatlarla karşılaştırmak suretiyle alım ve satım noktalarının üretildiği bölgeleri bulmaya çalışmaktır.

Yine fiyatların hareketli ortalama altına inmesi "sat", üzerine çıkması ise "al" sinyali anlamına gelmektedir. Bunun yanında bu yöntemi kullananlar açısından kısa vadeli olan hareketli ortalamanın uzun vadeli olan diğerinin üzerine çıkması "al", altına inmesi ise "sat" sinyali anlamındadır.

Örnek:

(Örnek olarak Exponential Moving Average seçilmiştir.)

```
var barDataModel = GetBarData();  
  
if (CrossAbove(barDataModel, ema, OHLCType.Close))  
{  
    SendMarketOrder(Symbol, BuyOrderQuantity, (OrderSide.Buy));  
    Debug("Alış emri gönderildi.");  
    Debug("Close = " + barData.BarData.Close);  
    Debug("Ema = " + ema.CurrentValue);  
}
```

```
}  
if (CrossBelow(barDataModel, ema, OHLCType.Close))  
{  
    SendMarketOrder(Symbol, SellOrderQuantity, (OrderSide.Sell));  
    Debug("Satış emri gönderildi.");  
    Debug("Close = " + barData.BarData.Close);  
    Debug("Ema = " + ema.CurrentValue);  
}
```

MSLIndicator(String, SymbolPeriod, OHLCType, Int32)

Moving Stoploss göstergesinin deęerini hesaplamak için kullanılır.



Kayı sınırlamak için kullanılan stop-loss özelliğinin, kazançla birlikte, stop loss seviyesini yukarı çekerek, daha sağlıklı bir stop-loss uygulaması haline getirilmiş halidir.

Örnek:

```
if (CrossBelow(msl, barData.BarData.Close))
{
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
    Debug("Stoploss seviyesinin altına inildi. Satış emri verildi.");
}
```

PDIIndicator(String, SymbolPeriod, OHLCType, Int32)

Price Direction Indicator (PDI) göstergesinin değerini hesaplamak için kullanılır. Yatırımcılar, fiyatların üst ve alt sıralarını belirlemek için Price Direction Indicator (PDI) kullanabilir. PDI, çoğu alım ve satım kombinasyonunun, fiyatlar tersine çevrildiğinde para kazandırma prensibiyle çalışır. Bir analiz periyodunun her bir günü için PDI değeri, analiz periyodunun ilk tarihinden o anki tarihe kadar tüm işlemler için kümülatif geri dönüşlere eklenen yüzdelik kazananlardır. PDI değerleri doğrudan hareketli ortalamalar gibi fiyatlardan elde edilmediğinden, bir fiyat tablosunda çizildiğinde PDI değerleri gerçek fiyatların çok altında veya üstünde konumlandırılabilir. PDI değerlerinin büyüklüğü önemli değildir; faydalı olan onların kalıplarıdır. Günlük veya haftalık kapanış fiyatları ve PDI değerleri grafiği, üstleri ve altları görmenize yardımcı olur. Fiyatlar yükselirken, PDI değerleri yükselme eğilimindedir ve fiyatlar düşerken, PDI değerleri düşme eğilimindedir. PDI her zaman fiyatlar düştükten sonra düşer. PDI değerindeki eğilimler fiyatların yönündeki değişiklikleri teyit etme eğilimindedir.

Örnek:

```
decimal temp;

if (temp != null)
{
    if (temp < pdi.CurrentValue)
    {
        SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
        Debug("Alış emri verildi.");
    }
    if (temp > pdi.CurrentValue)
    {
        SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
        Debug("Satış emri verildi.");
    }
    temp = pdi.CurrentValue;
}
else
{
    temp = pdi.CurrentValue;
}
```

** pdi.CurrentValue: Price Direction Indicator (PDI) göstergesinin o andaki değeri

RSIIndicator(String, SymbolPeriod, OHLCType, Int32)

Relative Strength Index göstergesinin değerini hesaplamak için kullanılır.



İlk olarak J.Welles Wilder tarafından kullanılan Göreceli Güç Endeksi (RSI), bir hisse senedinin incelenen periyot içindeki bir önceki güne göre yükselen günleriyle bir önceki güne göre düşen günlerini bularak bunları birbiri ile karşılaştırmak ve bu şekilde öngörülerde bulunabilmeyi amaçlamaktadır.

Bir önceki güne göre düşen ve çıkan günlerin birbiri ile mukayesesi esasına dayandığı için de bu bilgilerin kısa vadeli bilgiler olduğu ve bunların kısa süre içinde kullanılıp tüketilmesi gerektiği de açıktır.

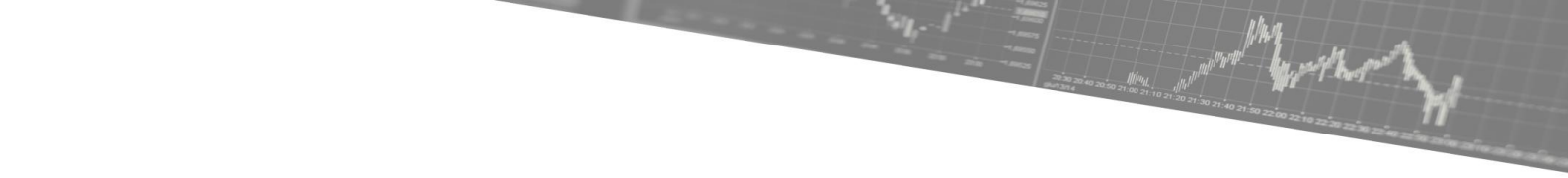
Her ne kadar vadeler kullanıcının tercihleri doğrultusunda belirlenecek olsa da RSI açısından vadelerin uzaması fiyat hareketlerine tepkisizleşme sorununu da beraberinde getirecektir.

RSI da genellikle 30 ve 70 referans değerleri kullanılmaktadır. Genel kabul 30 referans değerinin altına inen RSI değerlerinin aşırı satımı, 70 değerinin üzerine çıkan RSI değerlerinin de aşırı alımı işaret ettiği yönündedir. Diğer aşırı alım ve aşırı satım bölgeleri içeren göstergelerde olduğu gibi bu bölgelerde dolaşan RSI değerleri bize trendin yakın bir gelecekte sonlanabileceği sinyallerini vermektedir.

Örnek:

```
if (CrossAbove(rsi, DownLevel))
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));

if (CrossBelow(rsi, UpLevel))
    SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
```

- 
- ** rsi: Relative Strength Index göstergesinin o andaki değeri
 - ** DownLevel: Relative Strength Index göstergesinin alt çizgisinin değeri
 - ** UpLevel: Relative Strength Index göstergesinin üst çizgisinin değeri

SMAIndicator(String, SymbolPeriod, OHLCType, Int32)

Simple Moving Average (SMA) göstergesinin değerini hesaplamak için kullanılır.



SMA'lar genellikle trend yönünü belirlemek için kullanılır. SMA, yükseliyorsa yukarı yönlü trend, düşüyorsa aşağı yönlü trend vardır. 200 bar SMA uzun vadeli trend için ortak vekildir. 50 bar SMA'lar tipik olarak ara eğilimi ölçmek için kullanılır. Kısa dönem SMA'lar, kısa vadeli eğilimleri belirlemek için kullanılabilir.

SMA'lar genellikle fiyat verilerini ve teknik göstergeleri düzeltmek için kullanılır. SMA süresi ne kadar uzun olursa sonuç o kadar yumuşak olur, ancak SMA ile fiyat arasında daha fazla gecikme yaşanır.

SMA genellikle alım satım sinyallerini tetiklemek için kullanılır.

Örnek:

```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, sma, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Sma:" + sma.CurrentValue);
}

if (CrossBelow(barDataModel, sma, OHLCType.Close))
```

```
{  
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);  
    Debug("Satış Emri Gönderildi");  
    Debug("Close:" + barData.BarData.Close);  
    Debug("Sma:" + sma.CurrentValue);  
}
```

STDEVIndicator(String, SymbolPeriod, OHLCType, Int32)

Standart Deviation (Varyans'ın karekökü) göstergesinin değerini hesaplamak için kullanılır.



Senedin hareketliliğini ölçer. Sıfıra yaklaştıkça hareketlilik az demektir, 1'e yaklaştıkça hareketlilik yüksektir.

Bu indikatör tek başına teknik bir strateji belirlemede yeterli değildir. O nedenle diğer indikatörlerle birlikte kullanılır.

StochasticCCIIndicator(String, SymbolPeriod, OHLCType, Int32)

StochasticCCI göstergesinin deęerini hesaplamak için kullanılır.



StochasticCCI, paralel veya bitişik çubuklarda yukarı veya aşağı hareket ederse trendin devam edeceği anlamına gelir.

StochasticCCI'de mavi ok yeşil oku yukarı kestiği anda al, aşağı kestiği anda sat sinyali üretir.

StochasticFastIndicator(String, SymbolPeriod, OHLCType, Int32, Int32)

Stokastik Fast göstergesinin deęerini hesaplamak için kullanılır.



Periyodun içinde yer alan kapanışları periyottaki en yüksek ve en düşük değere olan uzaklıklarına göre deęerlendiren bir osilatördür.

Stochastic, periyodun içinde yer alan kapanışları periyottaki en yüksek ve en düşük değere olan yakınlıklarına göre deęerlendiren bir osilatördür. Daha doęru deyişle, periyot içindeki en yüksek ve en düşük seviyelere olan uzaklığına göre fiyatın deęerlendirilmesidir.

İlki %K ve ikincisi %D olarak isimlendirilen iki eğri ile gösterilen Stochastic'te %K, bugünün kapanışından periyodun en düşük gününün çıkarılması ile bulunan değerin, periyottaki en yüksek en düşük farkına bölünmesinin yüzdesel bir ifadesidir.

Ana eğrimiz olan %K 'nın bulunmasının ardından ikinci eğrimiz olan %D 'yi de %K 'nın hareketli ortalamasını alarak buluruz. George C. Lane tarafından geliştirilen bu gösterge için önerilen deęerler %K için 5 günlük bir periyot, bunun hareketli ortalaması için de 3 günlük bir basit ortalamadır. Bulduğumuz bu iki eğrinin aynı grafik üzerinde gösterilmesi ile elde ettiğimiz ise Fast Stochastic göstergesidir.

Ancak bu göstergenin her fiyat hareketine hemen tepki veren kararsız yapısı nedeniyle analistler yine bundan üretilen Slow Stochastic'i kullanmayı tercih ederler. Burada yapılacak olan ilk olarak %K 'nın hareketli ortalaması alınarak bulunan %D 'yi yavaşlatılmış %K değeri olarak kabul edip ana eğrinin yerine bunu koymaktır. Ardından bu yavaşlatılmış %K 'nın bir kez daha 3 günlük hareketli ortalaması alınarak yeni %D eğrisine ulaşılır. Bu yavaşlatılmış %K ve %D 'nin aynı grafikte gösterilmesi ile elde edilen gösterge ise Slow Stochastic olarak adlandırılır. Hem Fast Stochastic hem de Slow Stochastic de %K'nın %D'yi yukarı doğru kesmesi "al" sinyali olarak kabul edilirken, %K'nın %D'yi yukarıdan aşağıya keşişi ise "sat" sinyalini oluşturulmaktadır.

Stochastic 0'la 100 arasındaki değerlerden oluşur. Bu skala üzerine 20 ve 80 değerlerinden çizdiğimiz yatay çizgiler ise stochastic'in referans değerleridir. Diğer referans değeri kullanan göstergelerde olduğu gibi burada da bu değerlerin alt ve üst kısımları aşırı alım ve aşırı satım bölgeleri olarak adlandırılır. Bir senedin fiyatının 20 referans değerinin altına inmesi aşırı satıma ve bir süre sonra toparlanma olacağına, 80 referans değerinin üzerine çıkması ise aşırı alıma ve bir süre sonra gevşeme olacağına işaret eder.

Örnek:

```
if (CrossAbove(stochasticFast.StochasticFastK,
stochasticFast.StochasticFastD))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");

    Debug("stochasticFast.StochasticFastK:"+stochasticFast.StochasticFastK.CurrentValue);

    Debug("stochasticFast.StochasticFastD:"+stochasticFast.StochasticFastD.CurrentValue);
}
if (CrossBelow(stochasticFast.StochasticFastK,
stochasticFast.StochasticFastD))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");

    Debug("stochasticFast.StochasticFastK:"+stochasticFast.StochasticFastK.CurrentValue);

    Debug("stochasticFast.StochasticFastD:"+stochasticFast.StochasticFastD.CurrentValue);
}
```

```
** stochasticFast.StochasticFastK.CurrentValue: Stochastic Fast
göstergesinin %K değeri
** stochasticFast.StochasticFastD.CurrentValue: Stochastic Fast
göstergesinin %D değeri
```

StochasticRSIIndicator(String, SymbolPeriod, OHLCType, Int32)

Stochastic RSI göstergesinin değerini hesaplamak için kullanılır.

Stochastic ve RSI indikatörlerini birleştirir. Daha doğrusu RSI indikatörüne, Stochastic indikatörün uygulanması ile oluşmuştur. 0 ve 100 arasında değerler alır. 80 üstü aşırı alım ve 20 altı ise, aşırı satım işaretidir. Bu sınır değerler AL-SAT sinyalleri için kullanılabilir.

StochasticSlowIndicator(String, SymbolPeriod, OHLCType, Int32, Int32, Int32)

Yavaş Stokastik verilerini hesaplar. 2 ayrı çizgi oluşturmaktadır.



Bilgilendirme için yukarıda yer alan Stochastic Fast indikatörünün altındaki açıklamaya bakınız.

Örnek:

```
if (CrossAbove(stochasticSlow.StochasticSlowK,
stochasticSlow.StochasticSlowD)
                                && stochasticSlow.CurrentValue
<= DownLevel)
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
}
if (CrossBelow(stochasticSlow.StochasticSlowK,
stochasticSlow.StochasticSlowD)
                                &&
stochasticSlow.CurrentValue >= UpLevel)
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
}
```

```
** stochasticSlow. CurrentValue: Stochastic Slow göstergesinin o andaki
değeri
** stochasticSlow.StochasticSlowK: Stochastic Slow göstergesinin stochastic
slowK çizgisinin o andaki değeri
** stochasticSlow. StochasticSlowD: Stochastic Slow göstergesinin
stochastic slowD çizgisinin o andaki değeri
```

TMAIndicator(String, SymbolPeriod, OHLCType, Int32)

Triangular Moving Average(TMA), belirli sayıda veri noktası üzerinden ortalama fiyatı göstermesiyle diğer hareketli ortalamalara benzer. Bununla birlikte, üçgensel hareketli ortalama, çift düzgünleştirilmiş olması bakımından farklılık gösterir bu da iki kere ortalama anlamına gelir.



Üçgensel hareketli ortalamanın amacı, bir SMA kadar hızlı tepki vermeyen bir çizgi oluşturacak olan fiyat verisinin tahminini iki katına çıkarmaktır. TMA'yı ne için kullandığınıza bağlı olarak avantajlı veya dezavantajlı olabilir.

TMA değişken piyasa koşullarında hızlı bir şekilde tepki vermeyecektir; TMA'nın2ön değiştirmesi daha uzun sürecektir.

TMA gecikmeler olsa da faydalı olabilir. Fiyat ileri ve geri (aralık) hareket ederse, TMA o kadar fazla tepki vermeyecek, böylece eğilimin değişmediğini size bildirecek. TMA'nın yönünü değiştirmesine neden olmak için fiyatta daha kalıcı bir hareket gerekiyor.

Örnek:

```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, tma, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Tma:" + tma.CurrentValue);
}
if (CrossBelow(barDataModel, tma, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Tma:" + tma.CurrentValue);
}
```

** tma.CurrentValue: Triangular Moving Average göstergesinin o andaki değeri

TSFIndicator(String, SymbolPeriod, OHLCType, Int32)

Time Series Forecast göstergesinin değerini hesaplamak için kullanılır.



Time Series Forecast indikatörü Lineer Regresyon metodu kullanılarak hesaplanır. Lineer regresyon bir istatistik aracı olarak geçmiş değerleri karşılaştırarak gelecek fiyat değerlerini tahmin içindir. Bu amaçla trendlerin yukarıya veya aşağıya doğru meyillerini tanımlar ve bu sonuçları geleceğe taşır. Örneğin, fiyatlar yukarı doğru hareket ederken, TSF fiyatın yukarı meylini o anki fiyatla karşılaştırarak bu hesaplamayı geleceğe taşır.

Time Series Forecast, fiyatlar indikatörün altına düştüğünde trendi aşağı yönlü, indikatörün üstüne çıktığında ise yukarı yönlü kabul etmektedir. TSF İndikatörü eğer yönde ve eğimde bir değişiklik yoksa devam eden trendi tanımlar.

Örnek:

```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, tsf, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Tsf:" + tsf.CurrentValue);
}
if (CrossBelow(barDataModel, tsf, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Tsf:" + tsf.CurrentValue);
}
```

** tsf.CurrentValue: Time Series Forecast göstergesinin o andaki değeri

VMAIndicator(String, SymbolPeriod, OHLCType, Int32)

Variable Moving Average (VMA) Tushar S. Chande tarafından geliştirilmiştir. Variable Moving Average(VMA), yumuşatma sabitini piyasa volatilitesine göre ayarlayan üstel bir hareketli ortalamadır. Verilerin değişkenliği arttıkça, hassasiyeti artar. Bu üstel hareketli ortalamanın performansı, piyasa koşulları değiştikçe yumuşatma periyodunu ayarlamak için bir Volatility Index (VI) kullanılarak geliştirilmiştir.



Örnek:

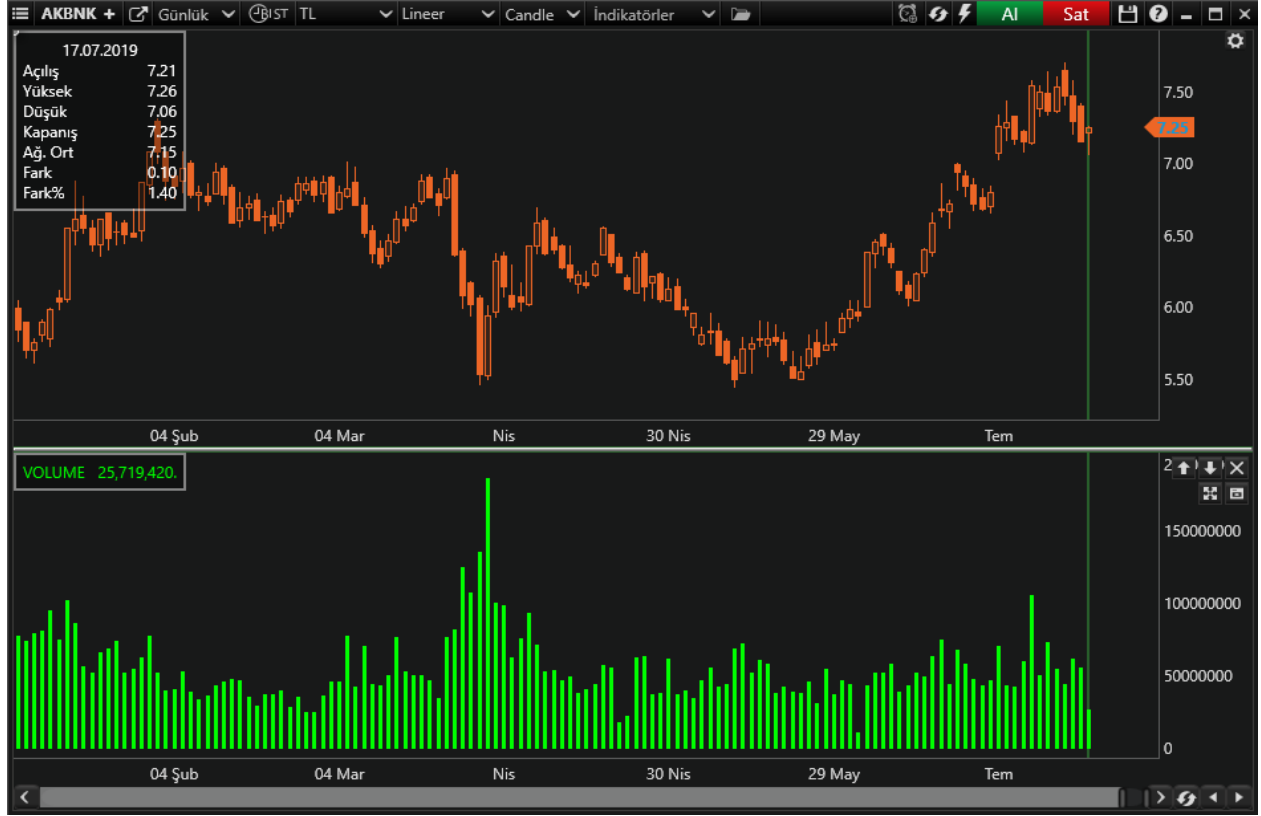
```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, vma, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Vma:" + vma.CurrentValue);
}
if (CrossBelow(barDataModel, vma, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Vma:" + vma.CurrentValue);
}
```

** vma.CurrentValue: Variable Moving Average göstergesinin o andaki değeri

VolumIndicator(String, SymbolPeriod)

Piyasada belirli bir süre zarfında ne kadar hisse senedinin el değiştirdiğini anlatan bir gösterge olan volume, genellikle fiyat hareketlerinin yorumlanması ve teyidinde kullanılmaktadır.




Yükselen bir piyasada yükselişe yüksek bir hacmin eşlik etmesi beklenirken düşüş genellikle hacimde bir daralmayı da beraberinde getirir. Bu yönüyle volume trendin dönüş yerlerinin tespitinde de yardımcı olabilmektedir. Şöyle ki, düşüşün sonlarına doğru iyice azalan işlem hacmindeki artış toparlanmanın başladığına işaret edebileceği gibi yükselişin sonlarına doğru azalmaya başlayan hacimde gevşemenin habercisi olabilmektedir.

Buradan hareketle, artmakta olan hacmin azalmaya başlaması ve belirli bir periyot boyunca azalan ya da sabit kalan bir hacmin artmaya başlaması, var olan trende ters yönlü bir gelişmenin olabileceğinin ilk sinyalleri olarak kabul edilebilir.

Destek direnç bölgelerinde de durum aynıdır. Bir destek ya da direncin kırılması sırasında artan işlem hacminin azalmaya başlaması bir başka destek ya da dirence yaklaşıldığı ve burada oyalanılacağı belki de dönüleceği anlamına gelirken, artışın aynı hızda devamı trendin aynı yönlü olarak devam edeceğinin bir belirtisidir.

Aynı zamanda yüksek işlem hacmi gerek yükselişlerde gerekse düşüşlerde olsun trendin aynı yönde devam edeceği kanaatinin yaygın olduğunu gösterir. Tersine düşük işlem hacimleri fiyatların dip oluşturduğu durumlar bir kenara bırakılacak olursa bir kararsızlık halini işaret eder.



Dow kuramında da üzerinde oldukça durulan volume için "trendi onaylamalıdır" denmektedir. Bunun kuramsal açıdan anlamı bir yükseliş trendi içinde yükselişler sırasında işlem hacminin artması gevşemelerde ise azalmasıdır. Tersine bir düşüş trendi içerisinde de düşüş sırasında işlem hacmi artmalı yükselişlerde azalmalıdır.

Bu indikatör tek başına teknik bir strateji belirlemede yeterli değildir. O nedenle diğer indikatörlerle birlikte kullanılır.

VolumeTLIndicator(String, SymbolPeriod)

VolumeTL(VOLT), fiyatlardaki deęişimin yönüyle işlem hacmini ilişkilendiren bir göstergedir. Basit bir şekilde senedin kapanışının bir önceki kapanışın üzerinde olmasıyla o gün oluşturulan işlem hacmi pozitif olarak işleme dahil edilirken, kapanışın bir önceki kapanışın altında kaldığı günlerde hacim negatif olarak hesaba katılmaktadır.

Genel kanı, VOLT'deki deęişimlerin fiyatlardaki deęişimlerden önce olduğu şeklindedir. Bunun ardında yatan varsayım ise diğer volume index göstergelerinde olduğu gibi fiyatlardaki salınımların ve hacmin azaldığı dönemlerde bilinçli yatırımcının alıma geçmesi ile VOLT'nin yükselmeye başlamasıdır. Fiyatların yükselişi ile senede ilgisi artan küçük yatırımcıların da gelmesiyle VOLT'deki artış hızlanacaktır.



VOLT'nin fiyatlarla uyum içinde olması beklenir. Yani VOLT'deki artışı fiyatlardaki artışın, düşüşü de fiyatlardaki düşüşün izlemesi gerekir. Gösterge ile fiyatlar arasındaki uyumsuzluklar trendin kısa bir süre sonra gösterge yönünde deęişeceği şeklinde yorumlanır.

Bir diğer uyumsuzluksa fiyatlardaki yükselişe VOLT'nin yeni tepelerle düşüşe de yeni diplerle eşlik edememesidir. Çünkü yükselen bir piyasada VOLT'deki yeni oluşan dip ve tepelerin bir önceki dip ve tepelerden daha yüksekte oluşması beklenirken, alçalan bir piyasada da VOLT'deki dip ve tepelerin bir önceki dip ve tepelerden daha aşağıda olması beklenir. Bunun gerçekleşmediği durumlar uyumsuzluk olarak kabul edilir.

Bu indikatör tek başına teknik bir strateji belirlemede yeterli değildir. O nedenle diğer indikatörlerle birlikte kullanılır.

WildersIndicator(String, SymbolPeriod, OHLCType, Int32)

Welles Wilder's Smoothing Average (WWS) J. Welles Wilder, Jr. tarafından geliştirilmiştir. Bu gösterge fiyat hareketlerini yumuşatır, yükseliş ve düşüş eğilimlerini tanımlamamıza ve belirlememize yardımcı olmak için kullanılır.



Fiyat hareketli ortalamanın üzerinde hareket ettiğinde, yükseliş sinyalleri, fiyat hareketli ortalamanın altına düştüğünde, düşüş sinyalleri verilir.

Örnek:

```
decimal data;
bool emir;

if (emir != null)
{
    if (data > wilders.CurrentValue)
    {
        if (emir == false)
        {
            SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Buy));
            Debug("Alış emri verildi.");
            Debug("Close:" + barData.BarData.Close);
            Debug("Wilders:" + wilders.CurrentValue);
        }
        emir = true;
    }
    if (data < wilders.CurrentValue)
    {
        if (emir == true)
        {
            SendMarketOrder(Symbol, OrderQuantity, (OrderSide.Sell));
            Debug("Satış emri verildi.");
            Debug("Close:" + barData.BarData.Close);
            Debug("Wilders:" + wilders.CurrentValue);
        }
        emir = false;
    }
}
else
{
    emir = true;
}
```

```
data = wilders.CurrentValue;
```

```
** wilders.CurrentValue: Wilders göstergesinin o andaki değeri
```

WMAIndicator(String, SymbolPeriod, OHLCType, Int32)

Ağırlıklı hareketli ortalamalar sinyalleri açısından basit hareketli ortalama ile aynıdır. Ayrıştığı nokta ise ortalamanın hesaplanmasında kullanılan yöntemdir. Basit hareketli ortalama uzun dönemlerde geçmiş hareketlerin ortalamaı etkileme sorunu ağırlıklı hareketli ortalama da giderilmeye çalışılmaktadır. Seçilen periyoda göre bugünkü fiyata daha çok önem vererek geçmişe doğru da önem derecesini azaltarak hesaplama gerçekleştirir.



Örneğin grafikte periyot 21 seçildiğinde bugünkü hareketin kapanış fiyatı 21 ile çarpılırken bir önceki hareketin kapanış fiyatı 20 ile çarpılır. 21 gün önceki hareketin kapanış fiyatına ise ağırlık verilmez. Böylece geçmiş hareketin daha yakın zamanı etkileme oranı aza indirgenmeye çalışılmaktadır.

Örnek:

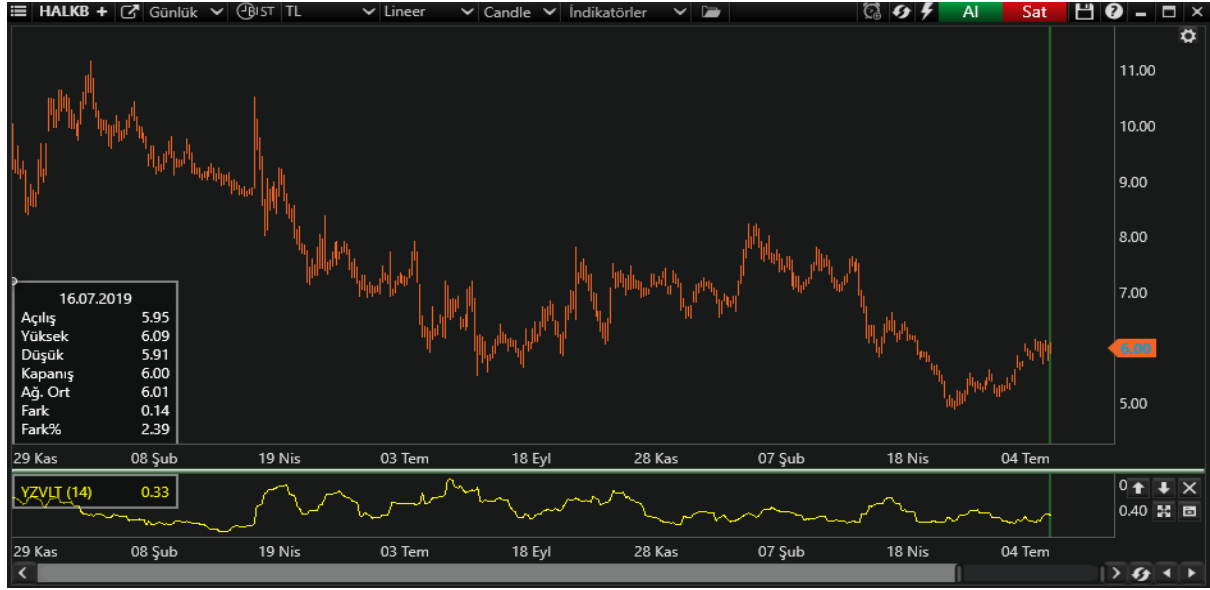
```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, wma, OHLCType.Close)
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Wma:" + wma.CurrentValue);
}
if (CrossBelow(barDataModel, wma, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Wma:" + wma.CurrentValue);
}
```

** wma.CurrentValue: Weighted Moving Average göstergesinin o andaki değeri

YZVLIIndicator(String, SymbolPeriod, OHLCType, Int32)

Yhang-Zhang Volatility göstergesinin deęerini hesaplamak için kullanılır.



Senedin açılışlardaki fiyat sıçramalarını da hesaba katarak volatilitte hesaplaması yapan bir indikatördür.

Bu indikatör tek başına teknik bir strateji belirlemede yeterli değildir. O nedenle diğer indikatörlerle birlikte kullanılır.

ZerolagIndicator(String, SymbolPeriod, OHLCType, Int32)

Zero lag exponential moving average göstergesi John Ehlers ve Ric Way tarafından geliştirildi. Amaç zaman içinde ortalama fiyat veren göstergelerin ardından tüm eğilim ile ilgili doğal gecikmeyi ortadan kaldırmaktır.



Buradaki fikir, normal üstel bir hareketli ortalama (EMA) hesaplaması yapmaktır ancak normal veriler üzerinde yapmak yerine gecikmiş bir veri üzerindedir. Veriler, günler öncesindeki 'gecikme' durumundan çıkarılır, böylece hareketli ortalamanın kümülatif etkisi ortadan kaldırılmaya (veya denenmeye) çalışılır.

Örnek:

```
var barDataModel = GetBarData();

if (CrossAbove(barDataModel, zeroLag, OHLCType.Close))
{
    SendMarketOrder(Symbol, BuyOrderQuantity, OrderSide.Buy);
    Debug("Alış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Zerolag:" + zeroLag.CurrentValue);
}
if (CrossBelow(barDataModel, zeroLag, OHLCType.Close))
{
    SendMarketOrder(Symbol, SellOrderQuantity, OrderSide.Sell);
    Debug("Satış Emri Gönderildi");
    Debug("Close:" + barData.BarData.Close);
    Debug("Zerolag:" + zeroLag.CurrentValue);
}
```

** zeroLag.CurrentValue: Zero lag Exponential Moving Average göstergesinin o andaki değeri