



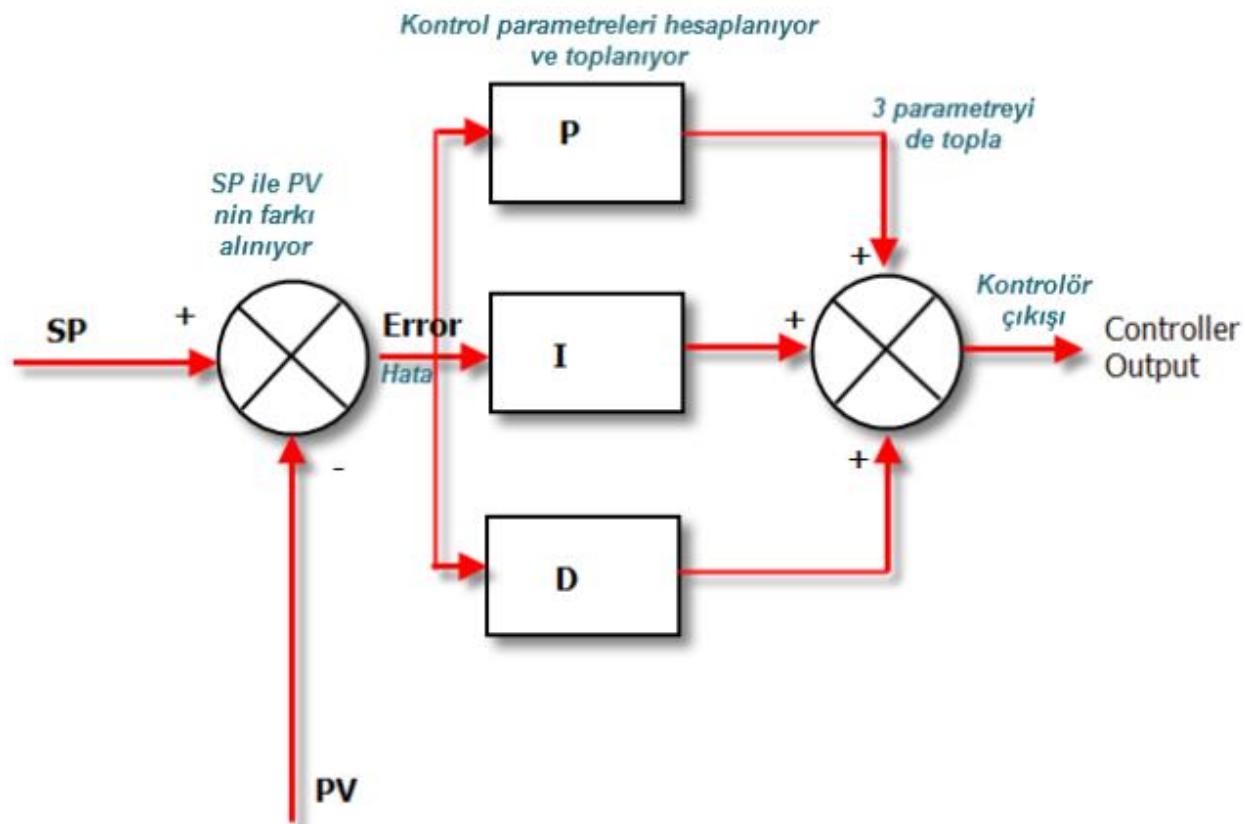
NJ Kontrolcüler için PIDAT Uygulaması

- PIDAT Fonksiyon Bloğu
- PIDAT Fonk. Bloğu Değişkenleri
- SSR Çıkışı

PIDAT Fonksiyon Bloğu

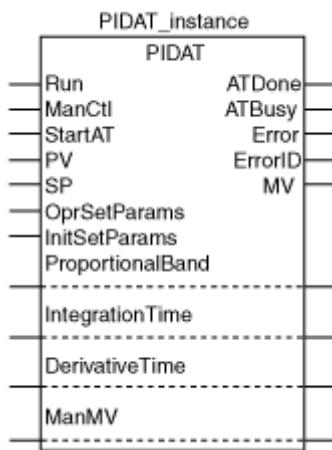
PID kontrol aşağıdaki blok şemada gösterildiği gibi ortam değerini ulaşımak istenilen değere

P,I,D parametreleri ile getirmek.



Şekil 1: PID Kontrol Blok Şeması

Sysmac Studio'da PIDAT ve PIDATHeatCool olmak üzere 2 fonksiyon bloğu yer almaktadır. PIDAT komutu autotuning ile PID kontrol gerçekleştirirken PDATHeatCool komutu autotuning ile ileri/geri(ısıtma/Soğutma) olmak üzere çift yönlü PID kontrolü gerçekleştirir. Şekil 2'de PID fonksiyon bloğunun görünümü yer almaktadır.



Şekil 2: PIDAT Fonksiyon Bloğu

PIDAT Fonksiyon Bloğu Açıklaması

PIDAT fonksiyon bloğunda yer alan (Şekil 2) Run girişi aktif edildiğinde komut çalışmaya başlar. Run girişi aktif olduğu sürece, PV (ortam sıcaklığı) oku, PID kontrol gerçekleştir ve MV(hesaplanan değer) çıkışına gönder, prosesi periyodik olarak gerçekleştirir. Run sinyali kesildiğinde PIDAT fonksiyon bloğu çalışmayı keser.

Autotuning ise sistem için optimal P,I,D değerlerini hesaplar. StartAT girişi aktif edildiğinde P,I,D değerleri hesaplanması için autotuning başlatılır.

Run

PID komutu Run girişi aktif olduğu sürece çalışır. Run girişi kesildiğinde PIDAT komutu çalışmayı durdurur.

ManCtl

ManCtl girişi PIDAT komutunun manuel mi otomatik mi çalışacağını seçmek için kullanılır. Eğer ManCtl girişi aktif ise operation *Manual* yani çıkışta yer alan MV değişkeni *ManMV* değişkeninde yer alan değer atanır.

StartAT

StartAT girişi P,I,D değerlerinin hesaplanması için autotuning’ı başlatır. *Run* girişi aktifken *StartAT* girişi aktif edilirse PID kontrolü bırakılır ve autotuning başlar. Eğer autotuning bitmeden StartAT girişi kesilirse autotuning iptal edilir.

PV

PV değeri kontrol edilen sistemin proses değeridir. (Örneğin ortam sıcaklığı, ortam basıncı vs.)

SP

Kontrol edilen sistemin ulaşması istenilen değer

OprSetParams.MVLowLmt&MVUpLmt

PID kontrolün sonunda hesaplanan MV değerine istenirse alt ve üst limit belirlemek için girişler kullanılabilir.

OprSetParams.ManResetVal

ManResetVal değişkeni P kontrol yaparken yani PV ve SP arasındaki fark 0 olduğunda MV değerini manual atamak içindir. Integral kontrol gerçekleştirilirken manual reset değişkeni dikkate alınmaz bu yüzden *ManResetVal* değeri IntegrationTime 0 iken aktiftir.

OprSetParams.MVTrackSw&MVTrackVal

MVTrackSw girişi dışarıdan bir giriş ile MV çıkışına *MVTrackVal* değerini atamak için kullanılan bir fonsiyondur. Bu giriş kesildiğinde ise MV değeri PID kontrolde hesaplanan MV değerine geri döner.

OprSetParams.StopMV

StopMV değeri *Run* girişi kesildiğinde MV çıkışının ne olması gerekiğinin yazıldığı bir değişkendir.

OprSetParams.ErrorMV

ErrorMV değeri hata durumunda MV çıkışının ne olması gerekiğinin yazıldığı bir değişkendir.

OprSetParams.Alpha

Alpha set point filtresi için kullanılan bir sabittir. Fabrika ayarlarında 0.65 atanmalıdır.

OprSetParams.ATCalcGain

ATCalcGain değeri autotuning esnasında yapılan hesaplama için kazanç ayarıdır. Eğer buradaki değer 1.00 olarak girilirse autotuning'de hesaplanan değerler aynen kullanılır. Kararlılığa öncelik vermek istenirse buradaki değer arttırılabilir, eğer tepkiye öncelik verilecekse *ATCalcGain* değeri azaltılabilir.

OprSetParams.ATHystrs

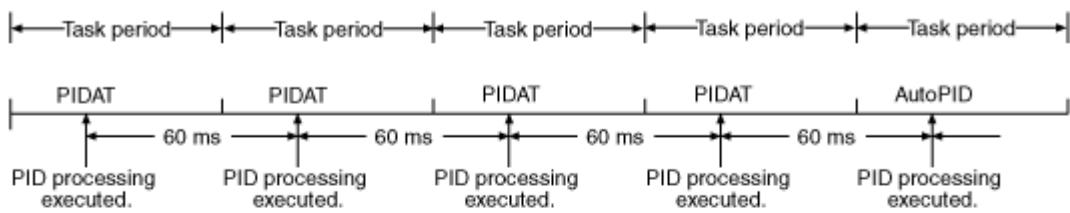
Autotuning'de kullanılan Hysteresis değeridir. Eğer *ATHystrs* küçük değerler ise daha karalı bir autotuning gerçekleştirilebilir ancak proses çok stabil değilse düzgün bir autotuning mümkün değilse bu değer artırılarak denenmelidir.

InitSetParams.SampTime

SampTime PID prosesinin uygulanacağı periyodun minimum değeridir. PID kontrol periyodik olarak tekrarlar. Eğer kullanıcı programının task periyodu *SampTime* değişkeninde belirtilen süreden uzun ise her task periyotta bir kere tekrarlar. Ancak *SampTime*'da belirtilen süre Task periyodundan büyükse, her *SampTime*'da belirtilen süreden sonraki task periyotta PID kontrol çalışır. (Bknz:Şekil 3)

Task period = 60 ms and *SampTime* < 60 ms

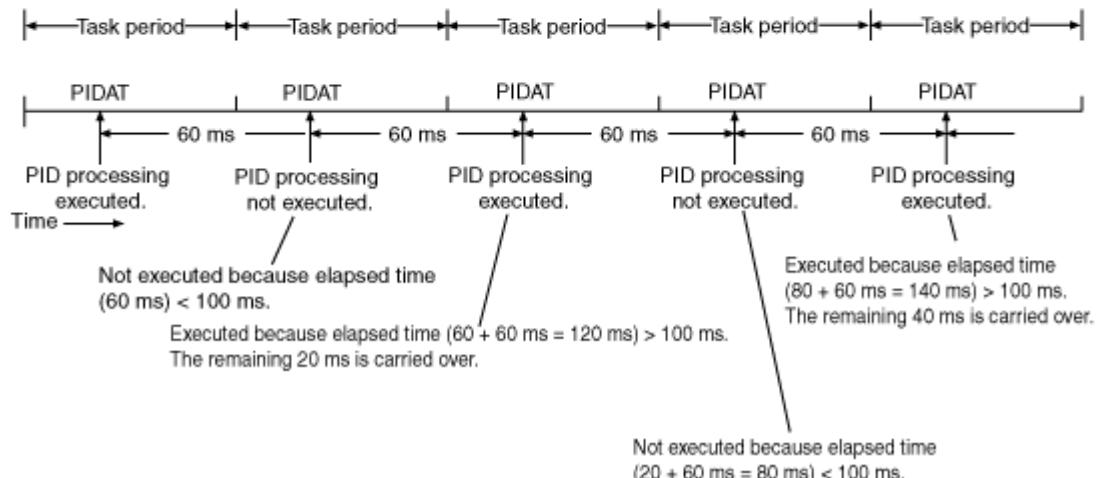
The task period is greater than or equal to *SampTime*, so PID processing is executed once every task period.



Time →

Task period = 60 ms and *SampTime* = 100 ms

The task period is less than *SampTime*, so DIP processing is not executed every period.



Şekil 3: SampTime'a göre PID tekrarı

InitSetParams.RngLowLmt&RngUpLmt

RngLowLmt&RngUpLmt PV ve SP için alt ve üst limit belirlenmek istenirse kullanılması içindir. Eğer PV veya SP 'ye gelen değer alt ya da üst limit dışı olursa hata meydana gelecektir.

InitSetParams.DirOpr

DirOpr değişkeni PV değerindeki değişime göre MV değerinin artacağını ya da azalacağını belirtmek için kullanılır. Örneğin sıcaklık kontrol için Forward Action soğutma olarak, Reverse Action ısıtma olarak çalışır.

ProportionalBand& IntegrationTime& DerivativeTime

PID sabitlerinden bir tanesidir. Autotuning ile hesaplanır.

SSR Çıkışı

Örnekte PIDAT komutu ile sıcaklık kontrolü gerçekleştirılmıştır. NX-TS kartından gelen sıcaklık değeri ortam sıcaklığı göstermektedir ve sıcaklık kontrolcüye giden NX-OD çıkışıdır ve SSR sürülmektedir. SSR kontrol edilmesi için PIDAT komutundan gelen MV değeri zamana orantılı (time proportional) çıkışa dönüştürülmelidir. Bunun için Sysmac Studio TPO(TimeProportionalOut) komutu kullanılarak çevrilmelidir. Ancak örnekte TimeProportionalOut komutu yerine Timer komutları ile yapılmıştır.

```
1 PulseCycTime:=TIME#1s;
2 ///////////////////////////////////////////////////////////////////
3
4 isitici_PulseOnTime:=MULTIME(PulseCycTime, isitici_MV/REAL#100.0);
5 isitici_TOF_instance(In:=BOOL#FALSE, PT:=isitici_PulseOnTime, Q=>isitici_Out);
6 isitici_TON_instance(In:=BOOL#TRUE, PT:=PulseCycTime, Q=>isitici_ResetPulse);
7 IF (isitici_ResetPulse=BOOL#TRUE) THEN
8     isitici_TOF_instance(In:=BOOL#TRUE);
9     isitici_TON_instance(In:=BOOL#FALSE);
10 END_IF;
11 IF ( (isitici_ATBusy=BOOL#TRUE) & (isitici_MV=REAL#100.0) ) THEN
12     isitici_Out:=BOOL#TRUE;
13 END_IF;
14
```

Şekil 4:TimeProportional çıkış