WHEN YOU NEED TO BE SURE

CyFlex® Pulse Width Modulation Control

Version 6

May 23, 2022

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	1/25/2019	Initial publication
2	8/23/2019	Format revisions
3	12/18/2019	Retrofit to new template
4	6/2/2021	Revised *Section 2 Starting PWM Control* on page 4 and *Section 3 Example PWM Specification File* on page 5
5	8/23/2021	Removed usage for `pwm_ctrl` and `pwm_specs` from *Section 2 Starting PWM Control* on page 4 and added hyperlinked cross-references to their respective cyflex.com usage help.
6	5/23/2022	Updated all hypertext linked cross-references to cyflex.com usage help descriptions

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.

  Example: Select the `cmdapp-`*`relVersion`*`-`*`buildVersion`*`.zip` file….

- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.

  Example: **Type**: Click **Select Type** to display drop-down menu options.

- Cross-references are designated in Arial italics.

  Example: Refer to *Figure 1*…

- Click intra-document cross-references and page references to display the stated destination.

  Example: Refer to *Section 1 Overview* on page 1.

  The clickable cross-references in the preceding example are *1*, *Overview*, and on page 1.

CyFlex Documentation

CyFlex documentation is available at https://cyflex.com/. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

List of Figures

1 Overview

The CyFlex Pulse Width Modulation (PWM) control task provides the ability to control an actuator using a PWM interface. This document describes:

- PWM Control concepts
- Configuring a PWM control specification file for use with an actuator
- Starting the PWM control task from go

Many types of actuators do not have proportional command signals. These actuators are often used in applications requiring precise positioning of the controlled element. A typical example is a servo-motor based actuator with only direction selection and ON/OFF control used to position an exhaust or inlet restriction valve.

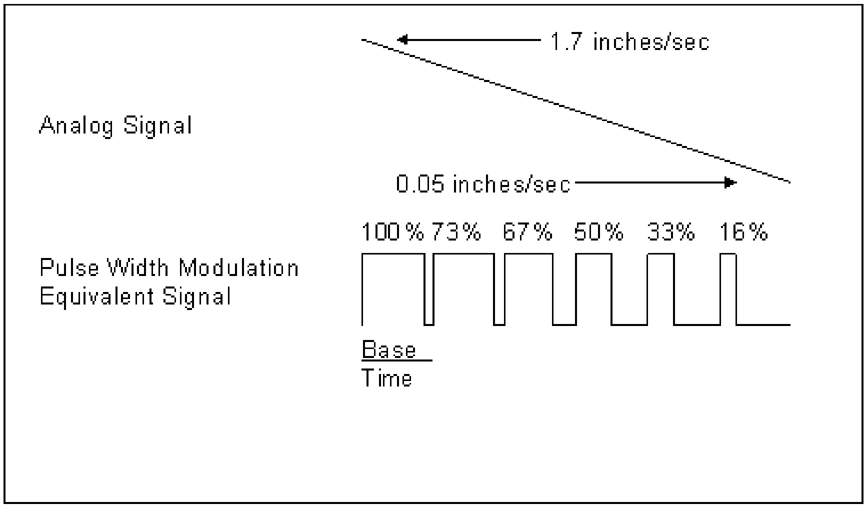
The lack of proportional control requires an alternate technique of control that is capable of achieving the required positioning accuracy. One such technique is Pulse Width Modulation (PWM) control, which uses a train of pulses with a fixed frequency and variable pulse width to control the rate at which the actuator will move. Refer to *Section 1.1 Concepts and Definitions* below.

CyFlex's PWM control task provides a mechanism for characterizing the actuator and how it is to be controlled. These pieces of information are placed in the PWM control specification file. Refer to *Section 3 Example PWM Specification File* on page 5.

Start one instance of the PWM control task for each PWM device to be controlled. The details of how to start the PWM control task are described in *Section 2 Starting PWM Control* on page 4.

1.1 Concepts and Definitions

Figure 1: PWM Conceptual Illustration



1.1.1 Pulse Width Modulation

This is a signal interface technique that uses a fixed frequency train of pulses with variable pulse widths to communicate an analog value. *Figure 1* on page 1 shows how a large analog signal is mapped to a longer pulse (or higher duty cycle) than a smaller signal.

1.1.2 Duty Cycle

The duty cycle is the ratio of pulse on-time to total time of a PWM signal. There is a linear relationship between duty cycle and the analog signal being represented by the PWM signal. In the above figure the 100% duty cycle corresponds to an actuator command value of 1.7 inches/sec and a 16% duty cycle corresponds to a 0.05 inches/sec duty cycle.

1.1.3 Slew Rate

The slew rate is the rate of change of the parameter affected by an actuator. This is the rate that would correspond to a 100% duty cycle. In the above example shown in *Figure 1* on page 1 the position of the actuator is being controlled so the slew rate is the displacement rate of the actuator. In the above example this corresponds to 1.7 inches/sec. In other PWM applications it may be a restriction pressure or temperature that is being controlled. In these cases, the slew rate would be in pressure/sec or temperature/sec units.

1.1.4 Minimum Duty Cycle Threshold

This is the minimum duty cycle that will result in a linear relationship between duty cycle and travel rate. Some devices require a minimum duty cycle due to non-linear components and pure delays, for example relay switching times, solenoid engagement times, brake disengagement times, motor static friction and deadband. In the above example shown in *Figure 1* on page 1, a minimum duty cycle of 16% is shown.

1.1.5 Feedback Limits

This is the range over which an actuator is capable of controlling the feedback parameter. Many PWM applications have physical limitations on the range of the controlled parameter. For some of these applications it is undesirable to operate the actuator beyond these limits as failure can result. For example, running a motor against a hard stop could result in the motor failing due to overheating. Actuators which are adversely effected by operating to their limits should have some form of position limiting. Actuator-integral position limit switches are sometimes available. The PWM control task can provide position limiting if an actuator position feedback signal is available to CyFlex and is the controlled parameter.

1.1.6 Base Time

The base time is the amount of time that corresponds to a 100% duty cycle PWM signal. The rate should be set at a value that is fast enough to provide the dynamic response and positioning accuracy required from the actuator but slow enough to ensure that the actuator will not be unable to move due to turn-on times of components. For example, if too long a base time period is selected then the actuator's command will be updated at a slower rate, resulting in overshoot or failure to accurately track a target. If too fast a base time period is selected then the actuator will be commanded to turn on and off again before the physical components are able to actually move. A good rule of thumb is to make the base time period roughly 20 times the minimum pulse period that provides a change in actuator displacement.

1.1.7 On-Command Deadband

This is the maximum amount of absolute error (between commanded setpoint and actual feedback) that will be tolerated in the actuator's position when an actuator is not moving. When the error exceeds this value when the actuator is not moving the actuator will be commanded to move. The on-command deadband should always be larger than the off-command deadband.

1.1.8 Off-Command Deadband

This is the amount of absolute error (between commanded setpoint and actual feedback) that will cause a moving actuator to stop moving. The off-command deadband should always be smaller than the on-command deadband.

1.1.9 Cycling

A cycle occurs each time an actuator is turned on or off. Some component's reliability and life are a function of the number of cycles experienced. Part of the process of tuning a PWM controlled actuator should involve minimizing the amount of cycling experienced by an actuator. Cycling can be reduced by having the maximum duty cycle set to 100% as this causes an actuator to remain energized across multiple base periods if operating at the maximum rate. For some systems the number of cycles should be tracked to permit preventative maintenance to be performed to avoid a failure.

1.1.10 On-Time

This is the total amount of time that an actuator is driven. This is the accumulation of the PWM pulses. Some components reliability and life are a function of the total on-time they experience. For these components it may be useful to track on-time in order allow preventative maintenance to be performed to avoid a failure.

1.1.11 Gain

The gain defines the relationship between duty cycle (or equivalent displacement rate) and error between commanded variable and actual feedback. A higher gain causes the actuator to move more quickly than a lower gain for the error. Higher gains may contribute to the inability to achieve the stability, resulting in continuously overshooting the desired setpoint.

2 Starting PWM Control

Refer to the following example command syntax to start the PWM control task in the `go` script:

```
pwm_ctrl DilFlow 8 SLO +c &
pwm_specs /specs/DilFlow.205
```

The preceding starts the PWM control task for the dilution flow actuator with a priority of 8 and a master scheduler process interval of the slow rate. The task is watchdog-critical. The PWM control task is then configured as specified in `/asset/specs/DilFlow.205`. The specifications may be modified and reactivated by reissuing the `pwm_specs` command with the specification path.

For related information, refer to cyflex.com usage help for:

- `pwm_ctrl`
- `pwm_specs`

3 Example PWM Specification File

The specification file for the `pwm_ctrl` task is called `<actuator>.nnn`. An example specification file for `DilFlow.205` is described below:

```
# PWM Control Specification File
# Device:  Mini Dilution Tunnel Dilution Air Flow Control Actuator
# Actuator Name
DilFlow

# Target Variable        Control Active
#                        Logical Variable
DilFlowTar               DilFlowCtrl
```

- `DilFlowTar` is a `REAL_VARIABLE` defined in `gen_labels.`*nnn*.
- `DilFlowCtrl` is a `LOGICAL_VARIABLE` defined in `gen_labels.`*nnn*.

```
# Direction              Motor Energize
# Logical Variable       Logical Variable
DilFlowDir               DilFlowActOn
```

- `DilFlowDir` and `DilFlowActOn` are `LOGICAL_VARIABLES` that may be defined in `do_specs.`*nnn* or `gen_labels.`*nnn*

```
# Gain Real Variable
# units – none
DilFlowGain

# Feedback     FeedbackLowerLimit     FeedbackUpperLimit
# Real Variable
DilFlow_VF     0.5[ft3/min]           25.0[ft3/min]
```

- The above limits are always a number units combination.
- `DilFlow_VF` is a `REAL_VARIABLE` that may be found in `inpt_specs.`*nnn* or `gen_labels.`*nnn*

```
# PWM_BaseTime
1[sec]

# MinRate   MinRateDutyCycle
.05[in]        16[%]
```

- A 5% duty cycle will move the actuator at 0.05 in/sec. The `/sec` is implied and should not be included in the specified units.

```
# MaxRate   MaxRateDutyCycle
1.7[in]        100[%]

# On Command DeadBand            Off Command DeadBand
# units - displacement           units - displacement
0.2[in]                           0.1[in]
```

- The actuator will turn on whenever the error exceeds 0.2 inches and turn off when the actuator moves to within 0.1 inches.

```
# NumCycles    TotalOnTime
DilFlowCyc     DilFlowOnT
```

- `DilFlowCyc` is an `INTEGER_VARIABLE` defined in `gen_labels.`*nnn*.
- `DilFlowOnT` is a `REAL_VARIABLE` defined in `gen_labels.`*nnn*.