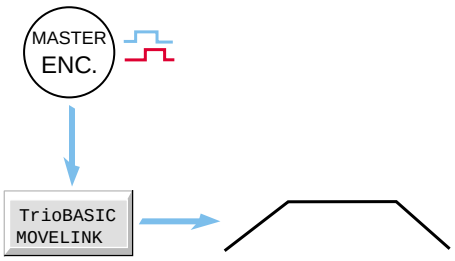
Unidirectional Software Gearboxes

The Problem

The software gearbox commands in TrioBASIC have an interesting consequence in that if the “input” - reference / master axis changes direction then the “output” servo axis will reverse too. Whilst in many cases this is not a problem, there are occasions when it is not acceptable for the servo axis to reverse and we need a software solution which prevents this occuring.

Considerations

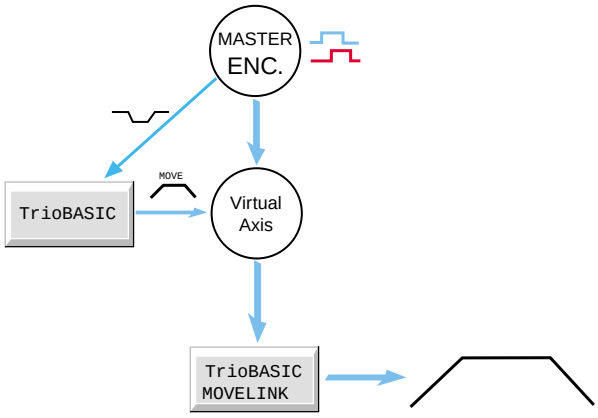
The ‘input’ to our software gearbox (**CONNECT, MOVELINK** or **CAMBOX**) is the measured position (**MPOS**) of the master axis, and this is normally a direct (in software) connection.



There is a scheme however where we can intercept the incoming encoder count and modify the value to cancel out negative moves before it is applied to the input of the software gearbox.

This is achieved by utilising a **virtual axis**, that is an axis which exists in software only. TrioBASIC has a useful command called **ADDAX** which allows us to superimpose the motion profile from one axis on top of another.

In this example we will add the measured position from the master encoder onto a virtual axis and then link the software gearbox, **MOVELINK** in this case, to the virtual axis.



Our method for cancelling out the negative move is quite simple. We continually track the change in measured position on the master encoder, providing the change between two samples is positive then we ignore it as we know that the axis is moving forward. If however the difference is a negative value then we know the axis must be moving in reverse. If we see a negative value then we immediately make a positive movement on the virtual axis of the same value. Because of the **ADDAX**, this will be summed with the incoming negative movement of the same magnitude, with a result of zero, effectively cancelling out the master signal and preventing movement on the slave axis.

So, we need to run a task specifically to monitor the master encoder and apply the changes, thus:

```
moves=0
virtual=3
enc=2

BASE(virtual)
ADDAX(enc)

old_mpos=MPOS AXIS(enc)
WA(1)
BASE(virtual)

loop:
    m=MPOS AXIS(enc)
    diff = m-old_mpos
    old_mpos = m

    IF diff<0 THEN MOVE(-diff)
    WAIT IDLE
GOTO loop
```

Then in our main run task we can simply apply the movelinks as normal:-

```
moves=0
virtual=3
enc=2

loop:
    MOVELINK(4000,40000,50,50,virtual) AXIS(moves)
GOTO loop
```

Although this example demonstrates the principle applied to a **MOVELINK** application, exactly the same principle should apply to a **CAMBOX** or **CONNECT.**