

# MPG-2314 MOVT

Ref No:ti2K-171207  
Last Modify 171213

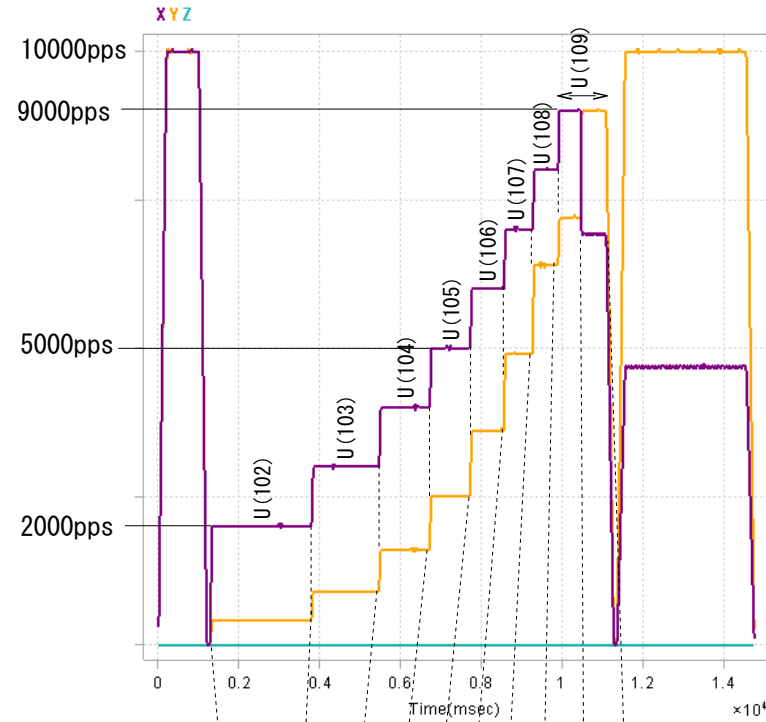
MOVTCOMMANDで2軸直線連続補間をしながら速度を変化させます。

MOVTCOMMANDは一つ先読みするので、速度変更などの処理を絡める場合、それを前にするか後にするかで動作が違います。ご注意ください。

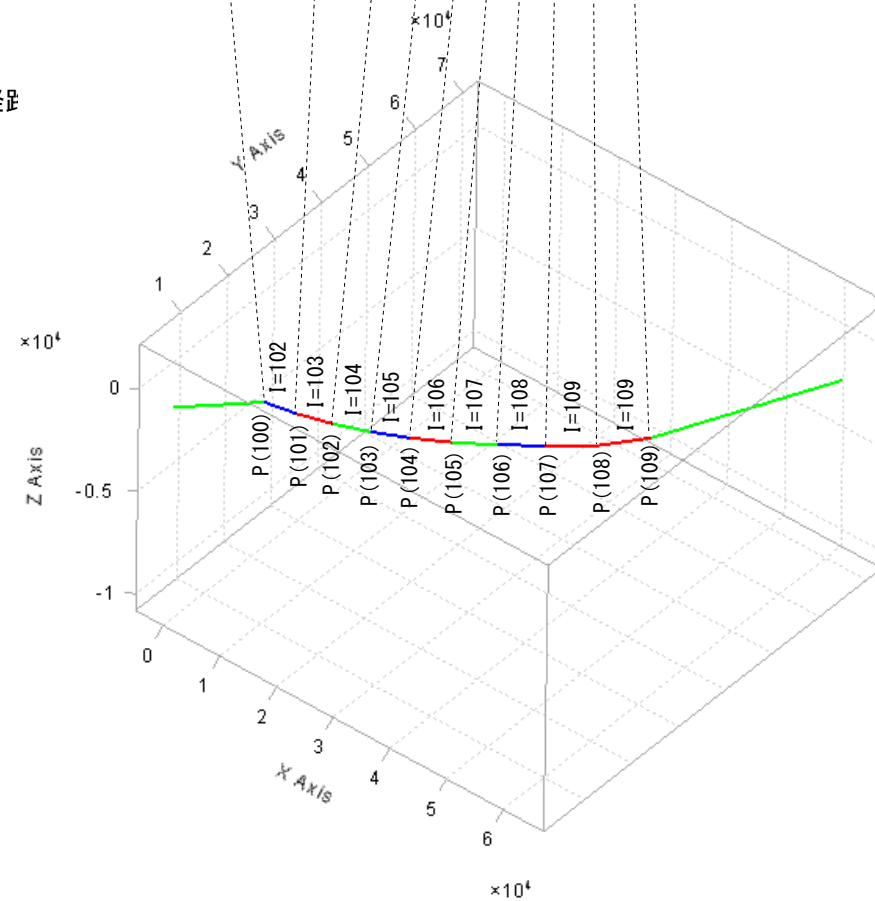
速度変更はSPEEDまたはFEEDコマンドで行います。ACCELコマンドは使わないで下さい。

[動作確認] MPC-2200X BL/I 1.14\_64 2017/10/20 , MPG-2314(CEP-128D,G)

## 速度変化



## 移動経路



## 速度変更と変数操作を MOVTCOMMAND の後で行う

```
CONST blue 1
CONST red 2
CONST green 3
PG 0
ACCEL X_A|Y_A 10000
FEED 100
CLRPOS
```

```
linecolor=green
MOVL 10000 10000 0 0 /* 起点へ移動
WAIT RR(ALL_A)==0
DS_DACL
TIME 100
```

```
FOR I=101 TO 109 /* 起点をP(100)とする
```

```
MOVTCOMMAND X_A|Y_A P(I) /* P(I)-P(I-1)の相対移動
```

```
SPEED X_A|Y_A U(I) /* 速度変更
```

```
linecolor=I%3+1
PR I linecolor
```

```
NEXT
```

```
EN_DACL
WAIT RR(ALL_A)==0
```

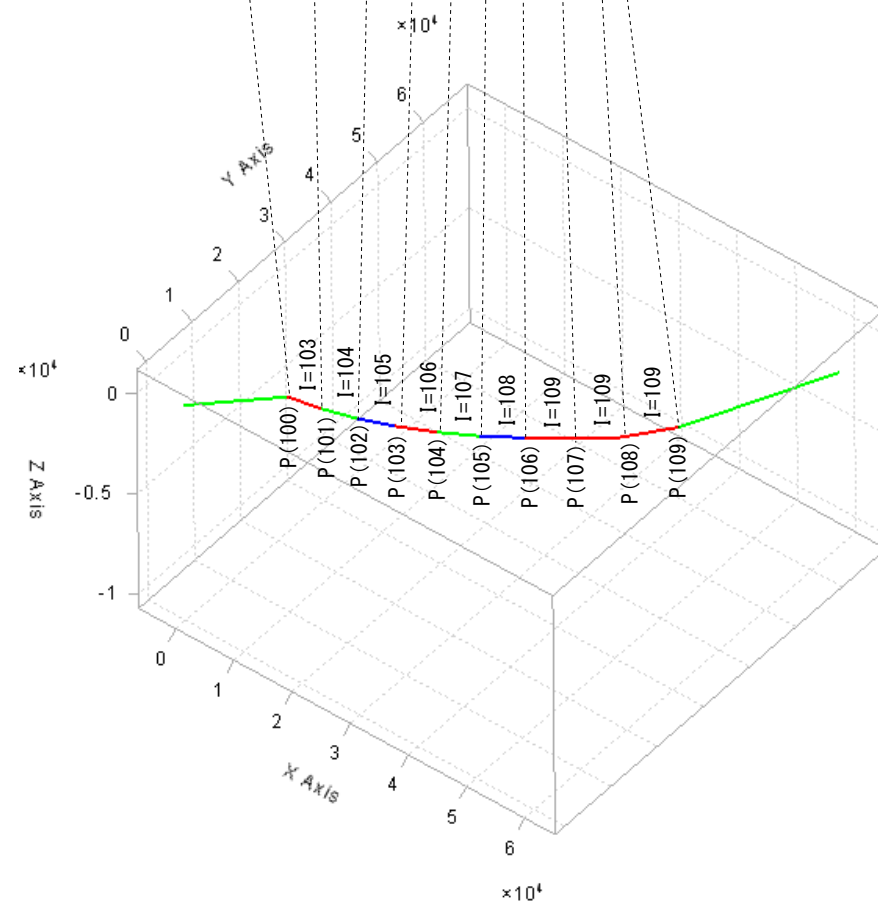
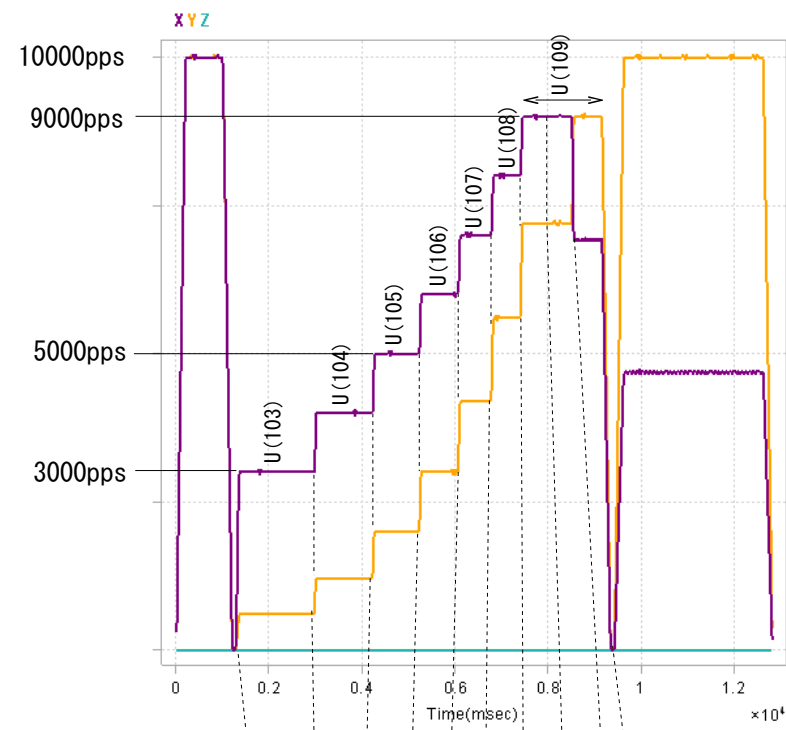
```
linecolor=I%3+1
PR I linecolor
TIME 100
ACCEL X_A|Y_A 10000
MOVL 70000 70000 0 0
WAIT RR(ALL_A)==0
```

## 点データ

X(100)=0	: Y(100)=0	: Z(100)=0
X(101)=5000	: Y(101)=1000	: Z(101)=0
X(102)=10000	: Y(102)=2500	: Z(102)=0
X(103)=15000	: Y(103)=4500	: Z(103)=0
X(104)=20000	: Y(104)=7000	: Z(104)=0
X(105)=25000	: Y(105)=10000	: Z(105)=0
X(106)=30000	: Y(106)=13500	: Z(106)=0
X(107)=35000	: Y(107)=17500	: Z(107)=0
X(108)=40000	: Y(108)=21500	: Z(108)=0
X(109)=45000	: Y(109)=28000	: Z(109)=0

## SPEED (pps)

```
U(100)=1000
U(101)=1000
U(102)=2000
U(103)=3000
U(104)=4000
U(105)=5000
U(106)=6000
U(107)=7000
U(108)=8000
U(109)=9000
```



速度変更と変数操作を MOVT の前で行う

```

CONST blue 1
CONST red 2
CONST green 3
PG 0
ACCEL X_A|Y_A 10000
FEED 100
CLRPOS

linecolor=green
MOVL 10000 10000 0 0 /* 起点へ移動
WAIT RR(ALL_A)==0
DS_DACL
TIME 100

FOR I=101 TO 109 /* 起点をP(100)とする

  linecolor=I%3+1
  PR I linecolor

  SPEED X_A|Y_A U(I) /* 速度変更

  MOVT X_A|Y_A P(I) /* P(I)-P(I-1)の相対移動

NEXT

EN_DACL
WAIT RR(ALL_A)==0

linecolor=I%3+1
PR I linecolor
TIME 100
ACCEL X_A|Y_A 10000
MOVL 70000 70000 0 0
WAIT RR(ALL_A)==0

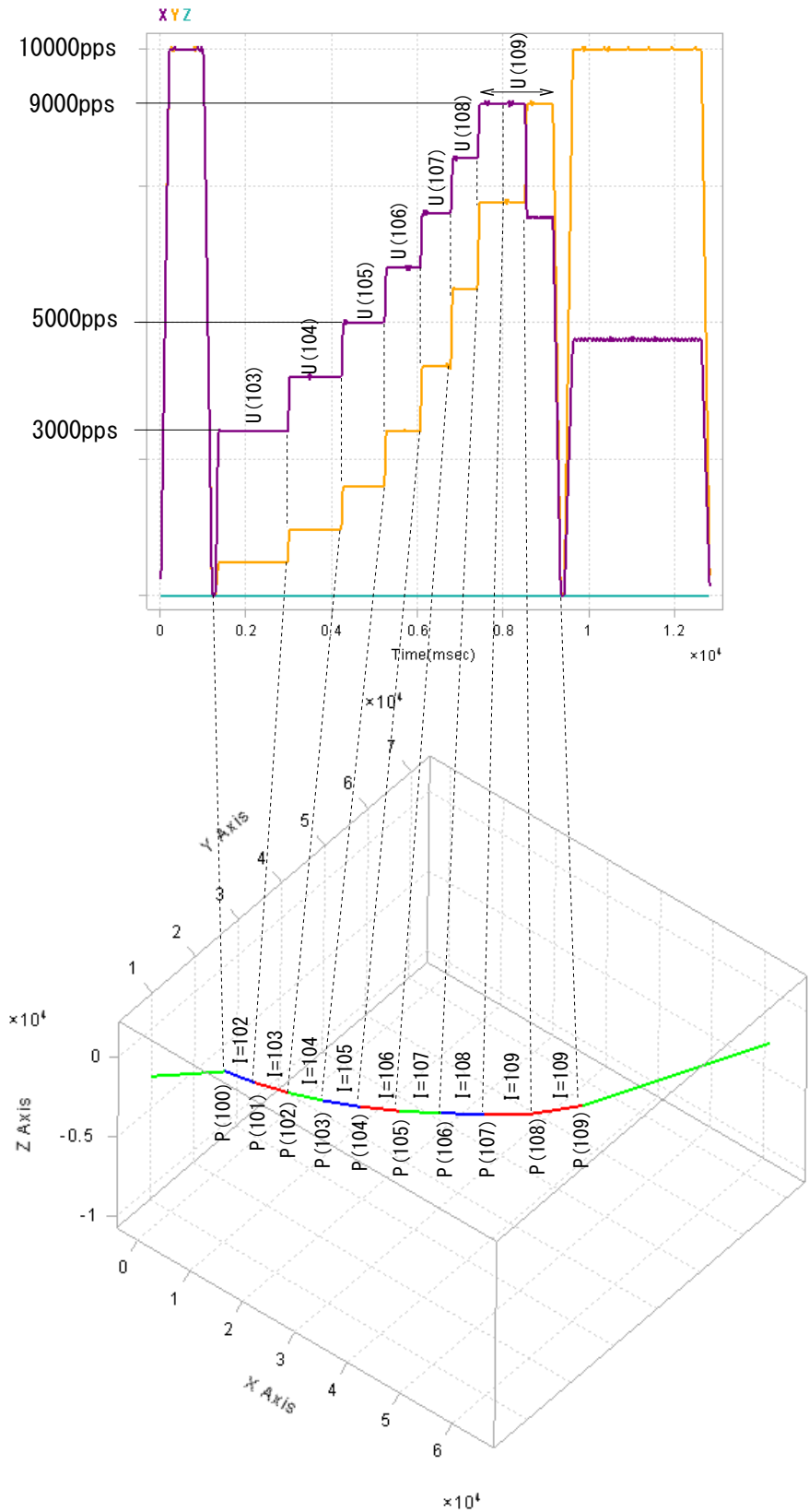
```

点データ

X(100)=0	: Y(100)=0	: Z(100)=0
X(101)=5000	: Y(101)=1000	: Z(101)=0
X(102)=10000	: Y(102)=2500	: Z(102)=0
X(103)=15000	: Y(103)=4500	: Z(103)=0
X(104)=20000	: Y(104)=7000	: Z(104)=0
X(105)=25000	: Y(105)=10000	: Z(105)=0
X(106)=30000	: Y(106)=13500	: Z(106)=0
X(107)=35000	: Y(107)=17500	: Z(107)=0
X(108)=40000	: Y(108)=21500	: Z(108)=0
X(109)=45000	: Y(109)=28000	: Z(109)=0

SPEED (pps)

U(100)=1000
U(101)=1000
U(102)=2000
U(103)=3000
U(104)=4000
U(105)=5000
U(106)=6000
U(107)=7000
U(108)=8000
U(109)=9000



速度変更を MOVT の前、変数操作をMOVTの後で行う

```

CONST blue 1
CONST red 2
CONST green 3
PG 0
ACCEL X_A|Y_A 10000
FEED 100
CLRPOS

linecolor=green
MOVL 10000 10000 0 0 /* 起点へ移動
WAIT RR(ALL_A)==0
DS_DACL
TIME 100

FOR I=101 TO 109 /* 起点をP(100)とする
  SPEED X_A|Y_A U(I) /* 速度変更
  MOVT X_A|Y_A P(I) /* P(I)-P(I-1)の相対移動

  linecolor=I%3+1
  PR I linecolor
NEXT

EN_DACL
WAIT RR(ALL_A)==0

linecolor=I%3+1
PR I linecolor
TIME 100
ACCEL X_A|Y_A 10000
MOVL 70000 70000 0 0
WAIT RR(ALL_A)==0

```

点データ

X(100)=0	: Y(100)=0	: Z(100)=0
X(101)=5000	: Y(101)=1000	: Z(101)=0
X(102)=10000	: Y(102)=2500	: Z(102)=0
X(103)=15000	: Y(103)=4500	: Z(103)=0
X(104)=20000	: Y(104)=7000	: Z(104)=0
X(105)=25000	: Y(105)=10000	: Z(105)=0
X(106)=30000	: Y(106)=13500	: Z(106)=0
X(107)=35000	: Y(107)=17500	: Z(107)=0
X(108)=40000	: Y(108)=21500	: Z(108)=0
X(109)=45000	: Y(109)=28000	: Z(109)=0

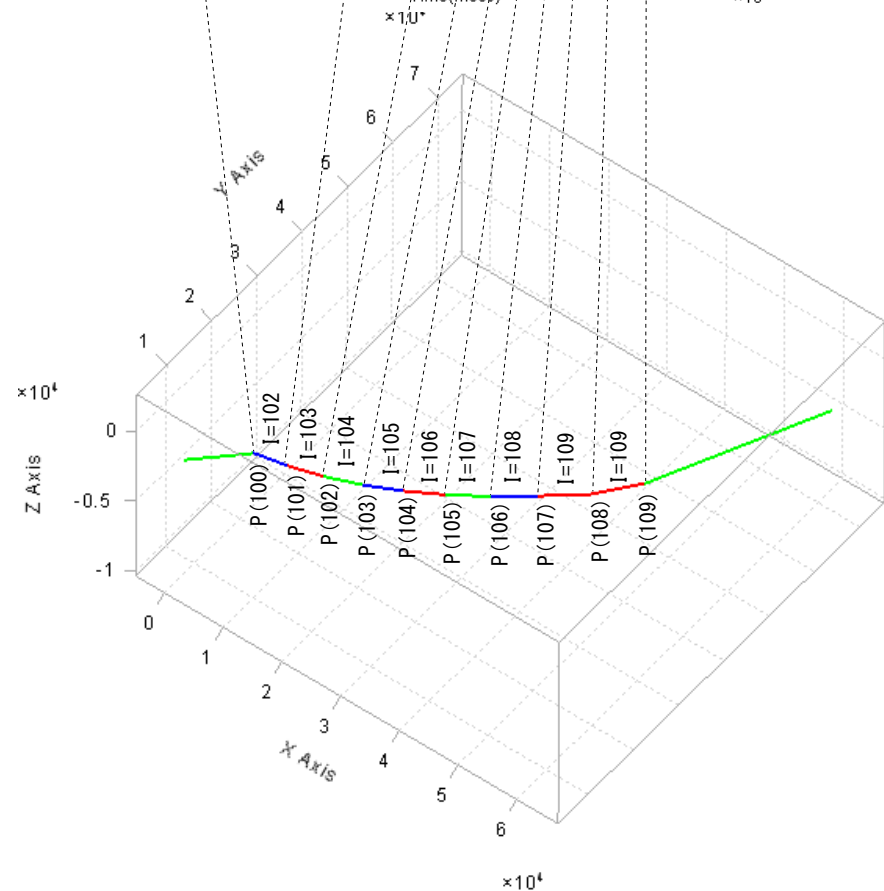
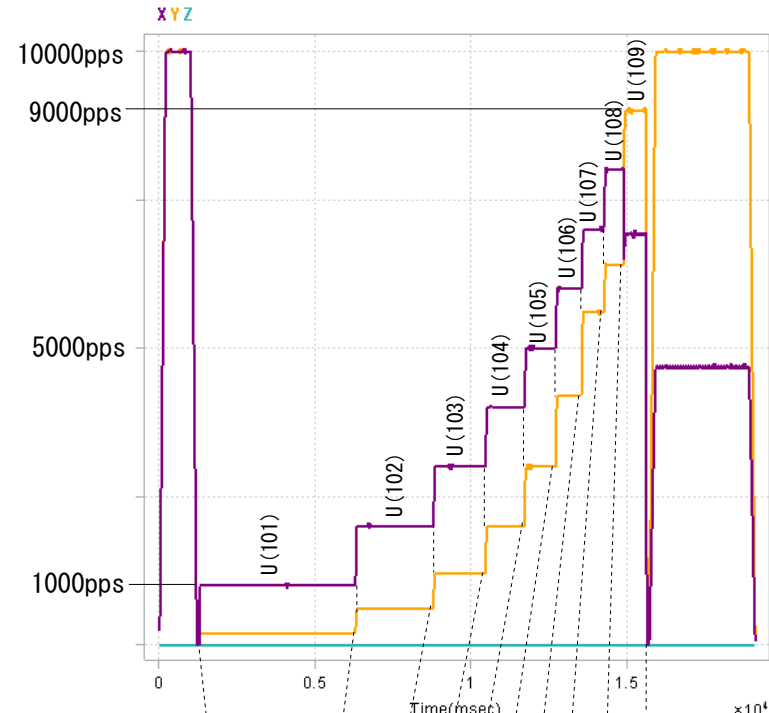
SPEED (pps)

U(100)=1000
U(101)=1000
U(102)=2000
U(103)=3000
U(104)=4000
U(105)=5000
U(106)=6000
U(107)=7000
U(108)=8000
U(109)=9000

# MPG-3514 MOVТ

MPG-3514は速度を指定してからMOVТを実行します。座標と速度を一緒に先読みするので、移動番号と速度番号は一致します。速度指定はSPEEDまたはFEEDコマンドで、必ずMOVТの前に行います。  
 [動作確認] MPC-2200X BL/I 1.14\_65 2017/12/08 , MPG-3514(CEP-174A)

速度変更を MOVТ の前、変数操作をMOVТの後で行う



```

CONST blue 1
CONST red 2
CONST green 3
PG 0
ACCEL X_A|Y_A 10000
FEED 100
CLRPOS

linecolor=green
MOVЛ 10000 10000 0 0 /* 起点へ移動
WAIT RR(ALL_A)==0
DS DACL
TIME 100

FOR I=101 TO 109 /* 起点をP(100)とする
    SPEED X_A|Y_A U(I) /* MOVТの前で速度変更
    MOVТ X_A|Y_A P(I) /* P(I)-P(I-1)の相対移動

    linecolor=I%3+1
    PR I linecolor

NEXT

EN DACL
WAIT RR(ALL_A)==0

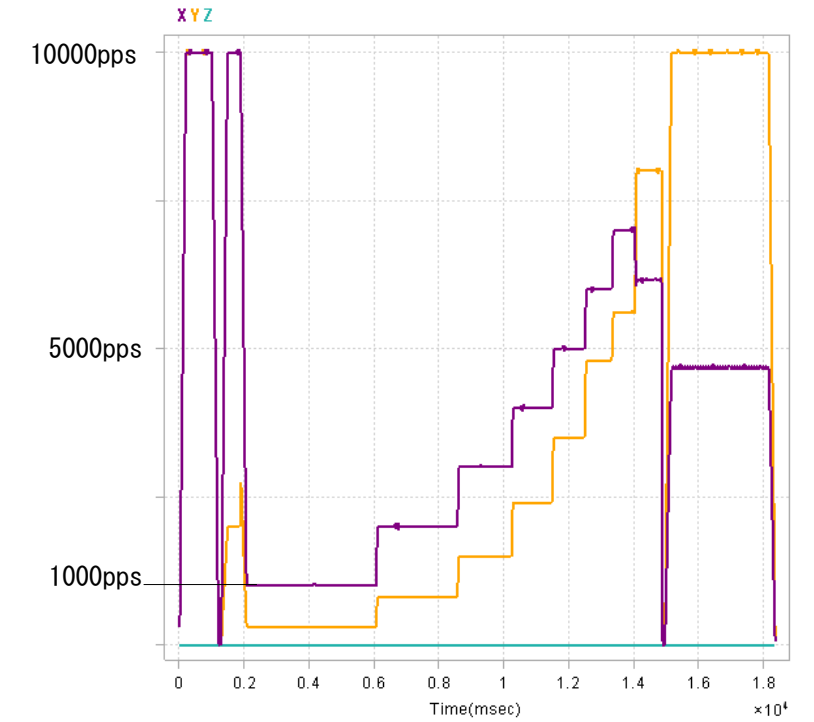
linecolor=I%3+1
PR I linecolor
TIME 100
ACCEL X_A|Y_A 10000
MOVЛ 70000 70000 0 0
WAIT RR(ALL_A)==0
    
```

点データ

X(100)=0	: Y(100)=0	: Z(100)=0
X(101)=5000	: Y(101)=1000	: Z(101)=0
X(102)=10000	: Y(102)=2500	: Z(102)=0
X(103)=15000	: Y(103)=4500	: Z(103)=0
X(104)=20000	: Y(104)=7000	: Z(104)=0
X(105)=25000	: Y(105)=10000	: Z(105)=0
X(106)=30000	: Y(106)=13500	: Z(106)=0
X(107)=35000	: Y(107)=17500	: Z(107)=0
X(108)=40000	: Y(108)=21500	: Z(108)=0
X(109)=45000	: Y(109)=28000	: Z(109)=0

SPEED (pps)

U(100)=1000
U(101)=1000
U(102)=2000
U(103)=3000
U(104)=4000
U(105)=5000
U(106)=6000
U(107)=7000
U(108)=8000
U(109)=9000



MOVТの後に速度変更すると正しく動作しません

```

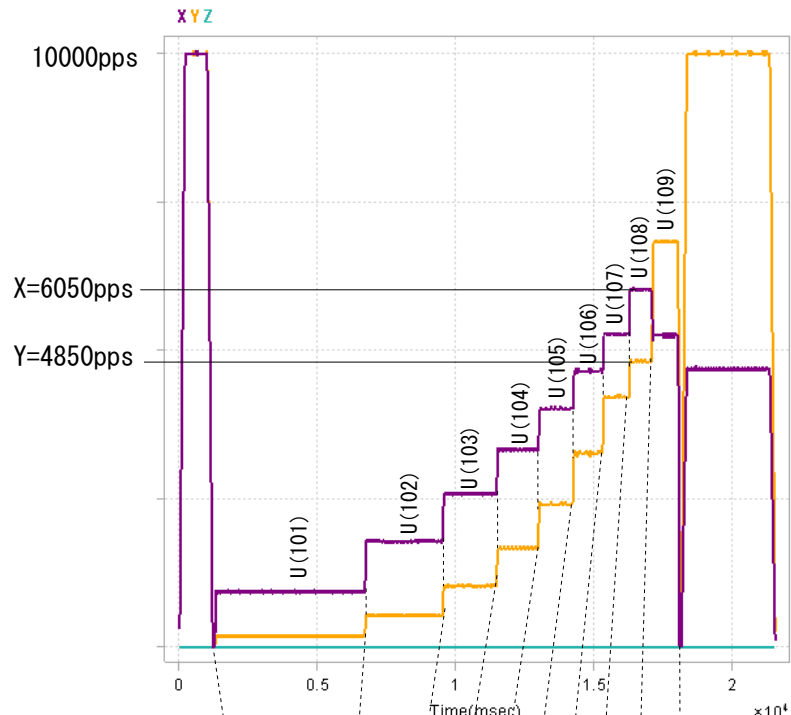
FOR I=101 TO 109 /* 起点をP(100)とする
    MOVТ X_A|Y_A P(I) /* P(I)-P(I-1)の相対移動
    SPEED X_A|Y_A U(I) /* MOVТの後の速度変更はNG

    linecolor=I%3+1
    PR I linecolor

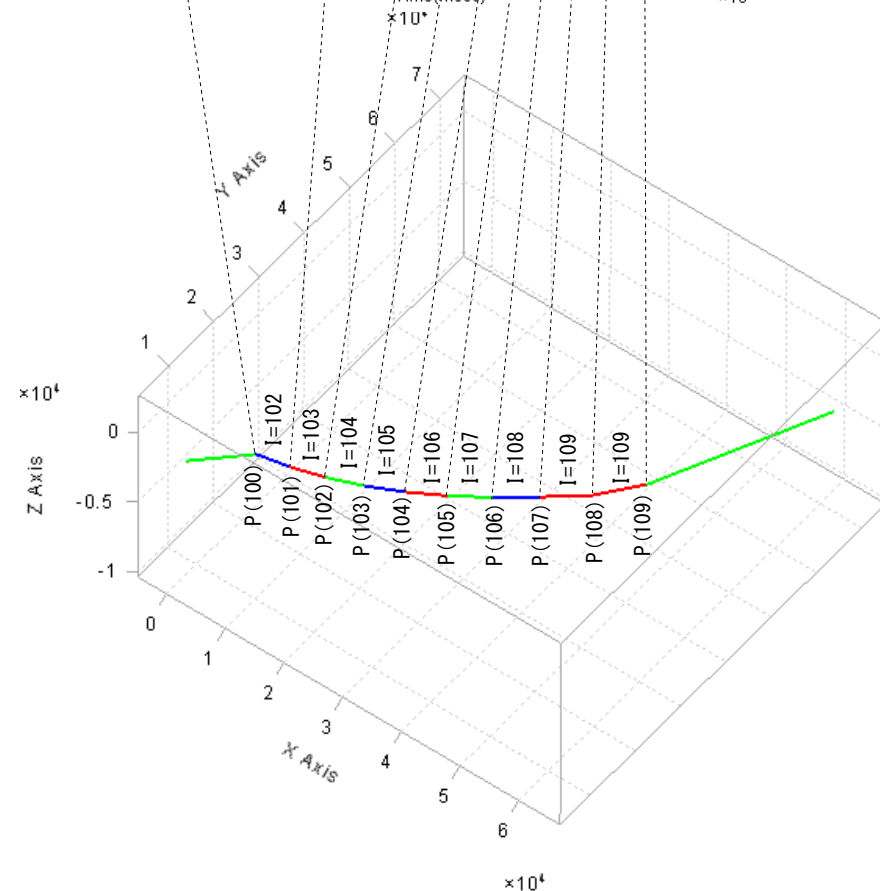
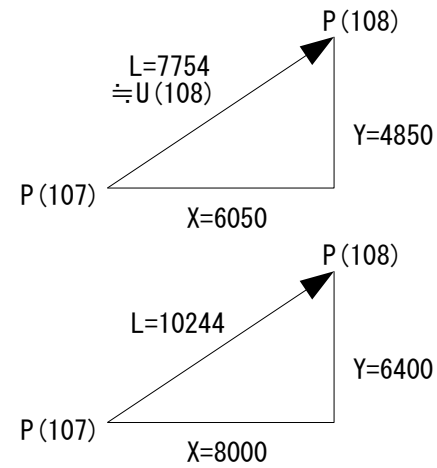
NEXT
    
```

直線連続補間 速度補正機能

ACCELのパラメータに  
 I\_ADJ2 (2軸直線補間 速度補正)  
 I\_ADJ3 (3軸直線補間 速度補正)  
 を追加する。(MPG-2314は無効)



速度補正機能  
 左図 P(107)→P(108)の場合、  
 XY合成速度がほぼU(108)になる。(上図)  
 補正がない場合は長軸にSPEED値  
 が適応される。(下図)



```

CONST blue 1
CONST red 2
CONST green 3
PG 0
ACCEL I_ADJ2|X_A|Y_A 10000 /* 2軸直線補間 速度補正
FEED 100
CLRPOS

linecolor=green
MOVL 10000 10000 0 0 /* 起点へ移動
WAIT RR(ALL_A)==0
DS_DACL
TIME 100

FOR I=101 TO 109 /* 起点をP(100)とする
    SPEED X_A|Y_A U(I) /* MOVJの前で速度変更
    MOVJ X_A|Y_A P(I) /* P(I)-P(I-1)の相対移動

    linecolor=I%3+1
    PR I linecolor

NEXT

EN_DACL
WAIT RR(ALL_A)==0

linecolor=I%3+1
PR I linecolor
TIME 100
ACCEL X_A|Y_A 10000
MOVL 70000 70000 0 0
WAIT RR(ALL_A)==0
    
```

点データ

X(100)=0	: Y(100)=0	: Z(100)=0
X(101)=5000	: Y(101)=1000	: Z(101)=0
X(102)=10000	: Y(102)=2500	: Z(102)=0
X(103)=15000	: Y(103)=4500	: Z(103)=0
X(104)=20000	: Y(104)=7000	: Z(104)=0
X(105)=25000	: Y(105)=10000	: Z(105)=0
X(106)=30000	: Y(106)=13500	: Z(106)=0
X(107)=35000	: Y(107)=17500	: Z(107)=0
X(108)=40000	: Y(108)=21500	: Z(108)=0
X(109)=45000	: Y(109)=28000	: Z(109)=0

SPEED (pps)

U(100)=1000
U(101)=1000
U(102)=2000
U(103)=3000
U(104)=4000
U(105)=5000
U(106)=6000
U(107)=7000
U(108)=8000
U(109)=9000