

位置決め完了と停止の繰り返しで、加工面がガタガタになってしまう。微い周期毎の速度計算をしているのがポイントである。これにより、滑らかな微いを実現している。また、演算誤差を防ぐため、100倍して計算して、端数の問題をクリアしている。このように、ある程度のテクニックを必要とするが、慣れればかなりの制御性や機能が実現できる。

図8は、この運転プログラムで実験した結果である。ギャップセンサを模擬して、信号発生器から正弦波を入力してX軸の微い動作を計測した。ほとんど遅れなく、滑らかに追従している。

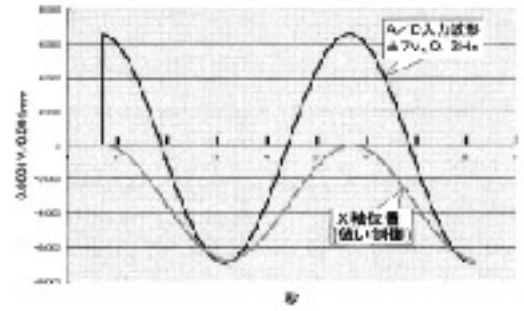


図8 微い動作の実験例

プログラムを作成してもらい、それを拡張していくのが効果的である。

1-5 マクロ機能の効果

簡単にまとめると以下の効果がある。

- a. 繰り返し計算などを利用して、運転プログラムのステップ数を少なくできる。
- b. ワーク種で変わる部分を変数化して、多様なワークに1本のプログラムで対応できる。
- c. その結果、運転プログラムの作成時間が短縮できる。
- d. ワーク種の追加や変更にも簡単に対応できる。
- e. 運転プログラムが読みやすい(保守しやすい)。

1-6 マクロ機能の導入の注意点

ソフトウェアのプログラミングに似たテクニックが多少必要である。そのため、サンプルをよく見るなど、ある程度の慣れも必要とする。また、ノウハウもあるので、初回はMCメーカーに相談して、それぞれの目的に応じたマクロ

2 マルチタスク

2つ以上の作業を同時に進行させるのがマルチタスクである。コンピュータやシーケンサの分野では、当たり前で汎用的であるが、MCの場合のマルチタスクは、マシンの応用や動作に直結するもので、その特徴をよく理解して導入すべきである。

MCメーカーや機種で詳細は異なるが、テクノのオープンMCの事例で説明する。

2-1 オープンMCのマルチタスク

(1) 2系列(マスター/スレーブ)運転

図9のように同時に2本の運転プログラムを実行できる。実際には、マスターとなる運転プログラムからスレーブのプログラムを起動する。これらの運転プログラムでは、起動のされ方以外はまったく同等で、すべての軸やI/Oの操作が可能である。ただ、同時に同じ軸に指令を行なうと実行エラーになる。

また、マシンの干渉を避けるため待ち合わせも必要である。一般には、作業ステーション毎に運転プログラムを作成するので、軸もグループに分かれるのが自然だが、ステーション間の搬送軸などは、両方の運転プログラムで指令できる方が便利である。

2ヘッドマシン、2台のマシンの制御、ロボットとローダの制御などの例が多い。

(2) バックグラウンドタスク

自動・手動モードによらず、常にバックグラウンドタスクは無限ループで実行されるので、機械周辺の制御や監視(油圧、材料供給など)に都合がよい。1命令に1制御周期を要するので、シーケンサに比べて処理能力は低いですが、簡単なI/O制御には、便利である。

図10では、油圧操作、コンベア操作、材料切れによるコンベア停止などを行なっている。

(3) その他のイベントタスク

運転中に例外(EXIT)、アラーム、リセットなどが発生したときに指定のタスクを起動して、それぞれに応じた動きが可能である。図11にその例を示す。

EXIT発生とは、テクノ言語のER/ENR□命令で汎用入力を判別した時の強制終了のことである。

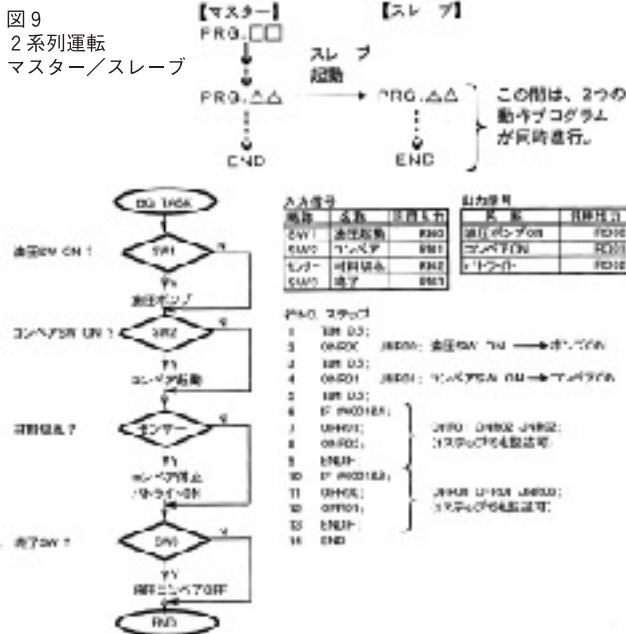


図10 バックグラウンドタスク(B.Gタスク)の例



図11 イベントタスクの例