

MCの「マクロ機能」と「マルチタスク」

(株)テクノ 山中 守

「緻密なサーボ制御」がマシンの生産性を高める基本要素であるのに対して、モーションコントローラ(MC)の「マクロ機能」や「マルチタスク」は、応用面で直接的に生産性を高める。また、機能の複合・集約・拡張性によりシステムの差別化やトータルのコストダウンを推進する。

1 マクロ機能とは

運転プログラムの中で、演算や条件判断をさせる機能がマクロ機能である。たとえば、MC内の状態や入力信号の状態に応じて、移動位置や動作手順などを変えることができる。

1-1 マクロ変数と演算の種類

MCのもっている情報のほとんどが「特定のNO.で指定する変数」として取り扱うことができる。表1にマクロ変数の例を示す。MCのステータス、パラメタ、座標値、入力信号などすべてが、運転プログラムで読み・書きできる(ステータスなどの一部は書き込み不可もある)。

表1 マクロ変数の例

変数NO.	分類名称	内 容
#1360	--- 座標値	32ビット座標値(100μm)
#1580	--- MOCステータス情報	#1590 --- MOCステータス、アラーム情報、オーバーライド値 プログラムNO.、実行ステップNO. など #1591 --- 各軸ステータス(詳細) #1592 --- 各軸アラーム (詳細)
#1580	--- 入出力情報	全ての入力信号、出力信号のON/OFF情報
#2360	--- サーボパラメタ	各軸のパラメタ全て
#2360	--- 工具長補正値	工具長補正データ
#4360	--- 位置情報	各軸位置(指令、検出、補正、誤)
#5360	--- 計画位置	各軸セト・サーボ位置情報(指令、アプリア)
#6360	--- 軌道入力	3次元方向4オクタンクグループで18進数で読み込み

表2 マクロ演算の種類

分類	内 容
代 入	変数△←△[変数] 数値[△] ← [△変数] △[変数] ← [数値]
判 別	IF △[変数] 即値[△] 実行[△変数] 即値[△] (1) --- (式の値が真の時) 実行 ELSE; (2) --- (偽の時) 実行 ENDIF
演 算	+ (加算) - (減算) * (乗算) / (除算) % (剰余) & (AND) (OR) ^ (XOR) < (左シフト) > (右シフト) = (等) > (大) < (小) == (等) >> (大) << (小) 条件が真→1 偽→0

[] --- 「」で区切られた内の、どれか1つを意味
「」 --- 省略可能 △ --- 「」の空白、又はタブ
変数 --- レジスタ指称 即値 --- NO.、-1などの数値

これらの変数に対して、表2のような四則演算、ビット演算、大小比較、真偽判別(IF/THEN)などを行ない運転状況を適切に変更できる。たとえば、ワークの種類、寸法などに応じて、軸の移動量や作業の内容を運転プログラムの中で自発的に変更できる。

マクロ機能を使うことで、運転プログラムの自由度と応用性は、格段に高まり、生産性は上がる。簡単な事例を紹介する。

1-2 カセットの取り出し

搬送ロボットなどでは、カセット(ワーク)の搬出や搬入の事例が多い。このとき、カセットはカセットストックに配列されているのが普通で、その取り出しや挿入の座標は、初期位置と繰り返し回数をもとに簡単に計算できる。

カセットは、たとえば図1のようにストックに配置されているとする。カセットと次のカセットの位置のオフセットは、d値として決まっており、順番に上からアクセスすれば、そのときの位置を繰り返し回数nで計算できる。図2は、その計算をしている運転プログラムである。

さらに、カセットの有無を検出するセンサを入力信号とすれば、搬出時に空きストックをパスしたり、挿入時に空いているストックを探して挿入することもできる。

1-3 リベットマシン

多様なワークのリベットかしめをするマシンである。ワークは、200種ほどあり、小ロットなためワークが混在し

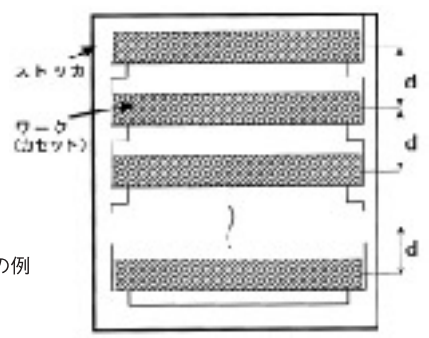


図1 カセットとストックの例

```

/* 初期処理 -----*/
#1036 = D; /* X軸座標 */
#1037 = D; /* Y軸座標 */
#1038 = D; /* Z軸座標 */
#1039 = -50[mm]; /* 軸補正 */
#1040 = 10; /* 軸速度n */
#1041 = D; /* 段差カウンタ */

/* ワーク取り出し処理 -----*/
M30R_PIC;
#1046 = #1403 & 1; /* ワーク取り出し要求番号 */
IF #1046 == 0; /* ワーク取り出し要求無し? */
JMF M30R_PIC; /* Yes -> 異常処理 */
ENDIF;

#1047 = #1403 * #1003; /* 軸座標位置算 */
#1048 = #1047 + #1002;
PTP #1048 #M001 Z#011; /* ワーク取り出し開始位置へ */

} /* 取り出しと搬送処理 */

#1049 = #1403 + 1; /* 次ワーク指定 */
IF #1049 < #1004; /* 全ワーク取り出し完了? */
JMF M30R_PIC; /* Yes -> 次ワーク処理 */
ENDIF;

/* 終了処理 -----*/
PTP #1036 #M001 Z#002; /* 各軸初期位置へ移動 */
END;
    
```

図2 運転プログラム