(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7294078号
(P7294078)

(45)発行日　令和5年6月20日(2023.6.20)

(24)登録日　令和5年6月12日(2023.6.12)

(51)国際特許分類
G05B 19/05 (2006.01)

FI
G05B 19/05 F

請求項の数　4　（全14頁）

(21)出願番号　特願2019-204825(P2019-204825)
(22)出願日　令和1年11月12日(2019.11.12)
(65)公開番号　特開2021-77219(P2021-77219A)
(43)公開日　令和3年5月20日(2021.5.20)
審査請求日　令和4年9月7日(2022.9.7)

(73)特許権者　000002945
オムロン株式会社
京都府京都市下京区塩小路通堀川東入南不動堂町８０１番地
(74)代理人　110002860
弁理士法人秀和特許事務所
(72)発明者　安田 圭
京都府京都市下京区塩小路通堀川東入南不動堂町８０１番地 オムロン株式会社内
(72)発明者　大谷 拓
京都府京都市下京区塩小路通堀川東入南不動堂町８０１番地 オムロン株式会社内
(72)発明者　関本 英彦
京都府京都市下京区塩小路通堀川東入南不動堂町８０１番地 オムロン株式会社内
最終頁に続く

(54)【発明の名称】　制御装置

(57)【特許請求の範囲】
【請求項１】
実行毎に全体が実行される第１の実行形式に従ったプログラムと、逐次実行される第２の実行形式に従ったプログラムと、を有する制御装置であって、
予め定められた制御周期毎に前記第１の実行形式に従ったプログラムを実行して第１の制御対象を制御するための第１の指令値を演算可能に構成された第１処理部と、
前記第２の実行形式に従ったプログラムを実行してインタプリタにより該プログラムの少なくとも一部が解釈されて生成される中間コードに従って前記制御周期毎に第２の制御対象を制御するための第２の指令値を演算可能に構成された第２処理部と、
前記第１の指令値および前記第２の指令値を前記制御周期毎に出力する出力部と、
を備え、
前記第１処理部と前記第２処理部のうちいずれか一方の処理部が、該第１処理部による演算と該第２処理部による演算の両方において使用可能な所定変数を保持し、該第１処理部と該第２処理部のうち他方の処理部は、該一方の処理部を介して該所定変数にアクセス可能に構成される、
制御装置。
【請求項２】
前記一方の処理部は前記第１処理部であり、前記他方の処理部は前記第２処理部である、
請求項１に記載の制御装置。
【請求項３】

前記所定変数は、前記一方の処理部による演算と前記他方の処理部による演算とにおいて、同一の変数名に設定され、
前記他方の処理部によって前記所定変数を使用するとき、該他方の処理部は、前記一方の処理部において該所定変数と同一の変数名を有する変数の値を該所定変数の値として使用する、
請求項１又は請求項２に記載の制御装置。
【請求項４】
前記一方の処理部による演算における前記所定変数の第１変数名と、前記他方の処理部による演算における前記所定変数の第２変数名とは異なり、
前記第１変数名と前記第２変数名との相関を画定する相関情報が保持され、
前記他方の処理部によって前記所定変数として前記第２変数名の変数を使用するとき、該他方の処理部は、前記相関情報に基づいて該第２変数名に関連付けられた前記第１変数名の変数の値を該所定変数の値として使用する、
請求項１又は請求項２に記載の制御装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、複数の制御対象を制御するための制御装置に関する。
【背景技術】
【０００２】
様々な生産現場において、ＰＬＣ（プログラマブルコントローラ）などの制御装置を用いたＦＡ（Factory Automation）技術が広く普及している。このような制御装置は、直接的に制御対象を制御するだけではなく、他の装置に制御指令を与えることで、間接的に制御対象を制御する場合もある。また、これまで複数の専用装置を用いて実現されていた制御システムを、より少ない数の制御装置に統合したいというニーズも生じている。例えば、特許文献１に示す技術では、ＰＬＣのＣＰＵユニットにおいて、モーション演算プログラムおよびユーザプログラムが同期して実行される。
【０００３】
また、特許文献２では、実行形式の異なる複数種類のプログラムに従う制御演算を単一の制御装置で実現する技術が開示されている。当該技術では、例えば、制御周期毎にプログラム全体が実行される形式のプログラムと、一部の解釈により生成される中間コードに従って逐次実行される形式のプログラムとに関して、それぞれの指令値演算を行い、制御周期毎に各指令値を併せて出力する構成が採用されている。
【先行技術文献】
【特許文献】
【０００４】
【文献】特開２０１２－１９４６６２号公報
特開２０１９－３６０４３号公報
【発明の概要】
【発明が解決しようとする課題】
【０００５】
複数の制御対象を制御するために、実行形式の異なる複数種類のプログラムに従う制御演算が行われる場合、それぞれの制御演算で使用される変数は、当該制御演算に応じて適宜設定されるのが一般的である。したがって、それぞれの制御演算が、異なる制御装置で行われる場合には、一方の制御装置による制御演算において、他方の制御装置が使用する変数にアクセスする必要があるときは、異なる制御装置間において当該変数にアクセスするための通信処理を行う必要がある。その結果、ユーザは、複数の制御対象を同時に制御するためには、各制御対象を制御するための制御演算を実行するプログラムを、その種類に従って作成する必要があるとともに、両プログラム（両制御装置）間での変数（データ）の通信に関する処理プログラムを作成する必要があった。

【０００６】
　また、複数の制御対象を制御するために、実行形式の異なる複数種類のプログラムに従う制御演算が単一の制御装置で行われる場合であっても、種類の異なるプログラム間での演算処理のための変数のやり取りについて、具体的な構成は十分に検討されていない。すなわち、従来技術においては、異なる種類のプログラムにおいて制御演算のための変数を共有する場合、どのように共有すべきかについては十分な検討が為されていない。そのため、異なる制御対象を単一の制御装置で行う場合であっても、ユーザによるプログラム生成の負荷は十分に軽減されていない。
【０００７】
　本発明は、このような問題に鑑みてなされたものであり、実行形式の異なる複数種類のプログラムに従う制御演算が単一の制御装置で行われる場合であっても、ユーザのプログラム生成の負荷を軽減し得る技術を提供することを目的とする。
【課題を解決するための手段】
【０００８】
　本発明においては、上記課題を解決するために、実行形式の異なる複数種類のプログラムの実行を単一の制御装置で可能とした上で、それぞれのプログラムで使用可能な変数を一方のプログラム側で保持し、他方のプログラムはその一方のプログラムを介して当該変数にアクセスする構成を採用した。当該構成の採用により、ユーザによる当該変数の管理が容易となり、ユーザのプログラム生成の負荷が軽減される。
【０００９】
　具体的に、本発明は、実行毎に全体が実行される第１の実行形式に従ったプログラムと、逐次実行される第２の実行形式に従ったプログラムと、を有する制御装置であって、予め定められた制御周期毎に前記第１の実行形式に従ったプログラムを実行して第１の制御対象を制御するための第１の指令値を演算可能に構成された第１処理部と、前記第２の実行形式に従ったプログラムを実行してインタプリタにより該プログラムの少なくとも一部が解釈されて生成される中間コードに従って前記制御周期毎に第２の制御対象を制御するための第２の指令値を演算可能に構成された第２処理部と、前記第１の指令値および前記第２の指令値を前記制御周期毎に出力する出力部と、を備える。そして、当該制御装置において、前記第１処理部と前記第２処理部のうちいずれか一方の処理部が、該第１処理部による演算と該第２処理部による演算の両方において使用可能な所定変数を保持し、該第１処理部と該第２処理部のうち他方の処理部は、該一方の処理部を介して該所定変数にアクセス可能に構成される。
【００１０】
　上記の制御装置は、実行形式の種類が異なるプログラムである、第１の実行形式に従ったプログラムと第２の実行形式に従ったプログラムを有している。前者の実行形式は、その実行毎にプログラム全体が実行される形式であり、後者の実行形式は逐次実行の形式である。このように異なる実行形式のプログラムが利用可能となることで、ユーザは、制御対象の制御に適したプログラムを適宜選択でき制御装置の利便性が高められる。
【００１１】
　そして、第１処理部が、制御周期毎に第１の実行形式に従ったプログラムを実行しそれに基づく第１の指令値を演算する。また、第２処理部は、インタプリタの解釈を経て生成される中間コードに従って制御周期毎に第２の実行形式に従ったプログラムに基づく第２の指令値を演算するように構成される。そして、出力部により第１の指令値と第２の指令値とが出力されることで、異なる実行形式のプログラムに基づいた各指令値を同期して出力することができる。
【００１２】
　このように構成される制御装置において、第１処理部と第２処理部のうちいずれか一方の処理部が、該第１処理部による演算と該第２処理部による演算の両方において使用される所定変数を保持するように構成される。すなわち、所定変数については両処理部が使用するものでありながら、一方の処理部のみが保持している。その上で、所定変数を保持し

ていない、第１処理部と該第２処理部のうち他方の処理部は、自己の演算に当該所定変数が必要な場合には、当該所定変数を保持している一方の処理部を介して、当該所定変数にアクセスする。この結果、他方の処理部も、当該所定変数を使用して自己の演算を実現できる。このとき、第１処理部と第２処理部は、単一の制御装置において形成されているため、ユーザは第１処理部と第２処理部との間のデータのやり取りを制御するプログラムを準備する必要は無く、ユーザの負荷軽減が図られる。

【００１３】

ここで、一例として、前記一方の処理部は前記第１処理部であり、前記他方の処理部は前記第２処理部であってもよい。また別法として、前記一方の処理部は前記第２処理部であり、前記他方の処理部は前記第１処理部であってもよい。

【００１４】

また、上述までの制御装置において、前記所定変数は、前記一方の処理部による演算と前記他方の処理部による演算とにおいて、同一の変数名に設定されてもよい。この場合、前記他方の処理部によって前記所定変数を使用するとき、該他方の処理部は、前記一方の処理部において該所定変数と同一の変数名を有する変数の値を該所定変数の値として使用してもよい。すなわち、所定変数について、一方の処理部と他方の処理部とで変数名を同一にすることで、その変数名を利用して両処理部間の所定変数のやり取りが行われる。

【００１５】

また、別法として、上述までの制御装置において、前記一方の処理部による演算における前記所定変数の第１変数名と、前記他方の処理部による演算における前記所定変数の第２変数名とは異なり、前記第１変数名と前記第２変数名との相関を画定する相関情報が保持されてもよい。この場合、前記他方の処理部によって前記所定変数として前記第２変数名の変数を使用するとき、該他方の処理部は、前記相関情報に基づいて該第２変数名に関連付けられた前記第１変数名の変数の値を該所定変数の値として使用してもよい。すなわち、所定変数について、一方の処理部と他方の処理部とで変数名は異なるものの変数名に関する相関情報が制御装置に保持されることで、その相関情報に基づいて両処理部間の所定変数のやり取りが行われる。

【発明の効果】

【００１６】

実行形式の異なる複数種類のプログラムに従う制御演算が単一の制御装置で行われる場合であっても、ユーザのプログラム生成の負荷を軽減し得る。

【図面の簡単な説明】

【００１７】

【図１】統合コントローラを含む制御システムの概略構成を示す図である。

【図２】統合コントローラの機能をイメージ化した機能ブロック図である。

【図３】統合コントローラで実行されるロボット制御において使用される変数とＰＬＣ制御において使用される変数の相関を示す図である。

【図４】統合コントローラで制御周期に従って実行される処理の流れを示す図である。

【図５】統合コントローラで実行されるロボット制御に関する第１のフローチャートである。

【図６】統合コントローラで実行されるロボット制御に関する第２のフローチャートである。

【発明を実施するための形態】

【００１８】

＜適用例＞

実施形態に係る制御装置の適用例について、図１及び図２に基づいて説明する。図１は、当該制御装置１００が適用される制御システム１の概略構成図であり、図２は、制御装置１００に形成される機能部をイメージ化した図である。

【００１９】

制御装置１００は、各種の設備や装置などの制御対象を制御する産業用コントローラに

相当する。制御装置１００は、後述するような制御演算を実行する一種のコンピュータである。制御装置１００は、フィールドネットワーク２を介して各種のフィールド機器と接続されてもよい。フィールド機器は、製造装置や生産ラインなど（以下、「フィールド」とも総称する。）に対して何らかの物理的な作用を与えるアクチュエータ、および、フィールドとの間で情報をやり取りする入出力装置などを含み、図１においては、フィールド機器として、ロボット２１０や、サーボドライバ２２０及びモータ２２２が例示されている。サーボドライバ２２０は、制御装置１００からの指令値（例えば、位置指令や速度指令など）に従って、モータ２２２を駆動する。また、ロボット２１０としては、パラレルロボット、スカラロボット、多関節ロボットが例示できる。このように制御装置１００は、ロボット２１０やサーボドライバ２２０及びモータ２２２を統合的に制御し得るように構成された制御装置であり、その詳細については後述する。

【００２０】

制御装置１００は、フィールドネットワーク２などを介して、１または複数のフィールド機器との間でデータをやり取りする。一般的に「フィールドネットワーク」は、「フィールドバス」とも称されるが、説明の簡素化のため、本願においては、「フィールドネットワーク」と総称する。制御装置１００は、各種のフィールド機器において収集または生成されたデータ（以下、「入力データ」とも称す。）を収集する処理（入力処理）、フィールド機器に対する指令などのデータ（以下、「出力データ」とも称す。）を生成する処理（演算処理）、生成した出力データを対象のフィールド機器へ送信する処理（出力処理）等を行う。

【００２１】

ここで、フィールドネットワーク２は、データの到達時間が保証される、定周期通信を行うバスまたはネットワークを採用することが好ましい。このような定周期通信を行うバスまたはネットワークとしては、ＥｔｈｅｒＣＡＴ（登録商標）等が知られている。そして、フィールドネットワーク２を介して、制御装置１００とフィールド機器との間でやり取りされるデータは、数１００μｓｅｃオーダ～数１０ｍｓｅｃオーダのごく短い周期で更新されることになる。なお、このようなやり取りされるデータの更新処理は、入出力リフレッシュ処理とも称される。

【００２２】

また、制御装置１００は、上位ネットワーク６を介して、他の装置にも接続されている。上位ネットワーク６には、一般的なネットワークプロトコルであるイーサネット（登録商標）やＥｔｈｅｒＮｅｔ／ＩＰ（登録商標）が採用されてもよい。より具体的には、上位ネットワーク６には、１または複数のサーバ装置１０が接続されてもよい。サーバ装置１０としては、データベースシステム、製造実行システム（ＭＥＳ：Manufacturing Execution System）などが想定される。製造実行システムは、制御対象の製造装置や設備からの情報を取得して、生産全体を監視および管理するものであり、オーダ情報、品質情報、出荷情報などを扱うこともできる。これに限らず、情報系サービスを提供する装置を上位ネットワーク６に接続するようにしてもよい。

【００２３】

ここで、図２に基づいて、制御装置１００の構成について説明する。制御装置１００は、上記の通り所定の制御演算を実行する一種のコンピュータであり、当該制御演算に必要なプロセッサやメモリを備えている。当該プロセッサは、ＣＰＵ（Central Processing Unit）、ＭＰＵ（Micro Processing Unit）、ＧＰＵ（Graphics Processing Unit）などで構成される。プロセッサとしては、複数のコアを有する構成を採用してもよいし、当該プロセッサを複数配置してもよい。当該メモリとしては、ＤＲＡＭ（Dynamic Random Access Memory）やＳＲＡＭ（Static Random Access Memory）などの揮発性記憶装置や、ＨＤＤ（Hard Disk Drive）やＳＳＤ（Solid State Drive）などの不揮発性記憶装置などで構成される。そして、プロセッサは、メモリに格納された各種プログラムを読出して実行することで、制御対象に応じた制御、および、後述するような各種処理を実現する。メモ

リには、基本的な機能を実現するためのシステムプログラムに加えて、制御対象の製造装置や設備に応じて作成されるユーザプログラム（ＩＥＣプログラム５１およびアプリケーションプログラム５２）が格納される。

【００２４】

なお、本願におけるＩＥＣプログラム５１とは、実行毎に全体がスキャンされて、実行毎に１または複数の指令値が演算されるプログラムであり、典型的には、国際電気標準会議（International Electrotechnical Commission：ＩＥＣ）によって規定された国際規格ＩＥＣ６１１３１－３に従って記述された１または複数の命令からなるプログラムを包含する。ＩＥＣプログラム５１には、シーケンス制御およびモーション制御の命令が含まれる。このようなＩＥＣプログラム５１は、制御周期毎にすべてのプログラムが実行（スキャン）される実行形式に対応するものであり、即時性および高速性が要求される制御に好適である。一方で、本願におけるアプリケーションプログラム５２は、ロボットを用いて特定の加工や動作を行うための制御プログラムであり、ロボットによる制御アプリケーションを実現するための１または複数の命令からなるプログラムを包含し、ＩＥＣプログラム５１とは区別される。ロボット制御に関するアプリケーションプログラム５２は、一例として、ロボット言語を用いて記述され、１行ずつ逐次実行されるインタプリタ方式が採用される。

【００２５】

そして、図２に示すように、制御装置１００は、制御アプリケーション処理部３０と、ＩＥＣプログラム処理部４０と、上位ネットワークインターフェイス２０と、下位ネットワークインターフェイス６０と、を有する。下位ネットワークインターフェイス６０は、ＩＥＣプログラム処理部４０および制御アプリケーション処理部３０と、フィールドネットワーク２を介して接続されているフィールド機器との間のデータのやり取りを仲介する。上位ネットワークインターフェイス２０は、ＩＥＣプログラム処理部４０および制御アプリケーション処理部３０と、上位ネットワーク６を介して接続されているサーバ装置１０との間のデータのやり取りを仲介する。例えば、制御装置１００は、上位ネットワーク６を介して接続されているサーバ装置１０から、生産の開始／終了といった指示を受ける。サーバ装置１０は、制御アプリケーションを動作させるためのアプリケーションプログラムおよびレシピ情報（生産に適したパラメータなどの情報）などを制御装置１００に送信することもある。

【００２６】

ＩＥＣプログラム処理部４０は、予め定められた制御周期毎にＩＥＣプログラム５１を実行（スキャン）して１または複数の指令値を演算する。すなわち、ＩＥＣプログラム処理部４０は、ＩＥＣプログラム５１に従って、制御周期毎に指令値を演算する。なお、本願では、モータ２２２を含んで構成される所定の装置を制御するためにＩＥＣプログラム５１の実行が行われるものとする。そして、モーション処理部４２は、ＩＥＣプログラム５１に含まれるモーション命令に従って、制御周期毎に指令値を演算する機能を提供する。すなわち、ＩＥＣプログラム５１に含まれるモーション命令は、複数の制御周期に亘る挙動を指示する命令（例えば、モータ２２２で構成される所定の装置の出力が何らかの軌道を描くための命令）を含む。このようなモーション命令が実行されると、実行されたモーション命令の指示内容に従って、モーション処理部４２が制御周期毎に指令値を演算する。すなわち、モーション処理部４２は、上記の所定の装置に対して制御周期毎に指令値を出力することで、モーション命令により指示された挙動を実現する。

【００２７】

また、ＩＥＣプログラム処理部４０は、共有メモリ４１を有している。共有メモリ４１は、モーション処理部４２による指令値の演算処理に使用される所定の変数に関するデータ（変数データ）を保持している。したがって、ＩＥＣプログラム処理部４０は、所定の装置の挙動を制御するための指令値演算に際して、適時、共有メモリ４１へのアクセスを行う。また、この共有メモリ４１に保持されている変数データは、後述する制御アプリケーション処理部３０からもアクセス可能に構成されており、その詳細については後述する

。また、共有メモリ４１には、上述した指令値演算のための変数データ以外のデータが格納されてもよい。例えば、本願開示では、ＩＥＣプログラム処理部４０による処理結果の一部または全部が共有メモリ４１に格納され、制御アプリケーション処理部３０は、ＩＥＣプログラム処理部４０の共有メモリ４１に格納されたデータを参照することができる。また、制御アプリケーション処理部３０からＩＥＣプログラム処理部４０の共有メモリ４１へのデータ書込みも可能であってもよく、このように制御アプリケーション処理部３０から書込まれたデータは、モーション処理部４２から参照可能とされる。

【００２８】

次に、制御アプリケーション処理部３０は、アプリケーションプログラム５２およびレシピ情報などに基づいて、制御アプリケーションを制御するための指令値を演算する。なお、本願では、ロボット２１０を制御するためにアプリケーションプログラム５２が制御アプリケーション処理部３０によって実行されるものとする。制御アプリケーション処理部３０は、ＩＥＣプログラム処理部４０による指令値の演算および出力と同期して、制御アプリケーション用の指令値を演算および出力する。すなわち、制御アプリケーション処理部３０は、ＩＥＣプログラム処理部４０による演算処理と同期して、ロボット２１０を制御するための指令値の演算処理を実行する。なお、このＩＥＣプログラム処理部４０と制御アプリケーション処理部３０の同期処理については、後述する。ＩＥＣプログラム処理部４０による指令値の演算処理と同期した指令値の演算を実現するために、制御アプリケーション処理部３０は、モーション処理部３２と、バッファ３３と、インタプリタ３４とを含む。

【００２９】

インタプリタ３４は、逐次アプリケーションプログラム５２の少なくとも一部を解釈して中間コードを生成するとともに、生成した中間コードを格納するバッファ３３を有する。本願における中間コードは、制御周期毎に指令値を演算するための命令を包含する概念であり、１または複数の命令、あるいは、１または複数の関数を含んでもよい。そして、モーション処理部３２は、インタプリタ３４が事前に生成しバッファ３３に格納されている中間コードに従って、制御周期毎に指令値を演算する。一般的に、アプリケーションプログラム５２に記述される命令（コード）は、逐次実行されるために、指令値の演算周期を保証できないが、本願開示では、このように中間コードを利用することで、モーション処理部３２は、制御周期毎に指令値を演算できるようになる。中間コードにおいて記述される命令には、各制御アプリケーションに応じた座標系が用いられてもよい。

【００３０】

また、制御アプリケーション処理部３０は、必要に応じて、モーション処理部３２による、ロボット２１０の制御のための指令値の演算処理のために、適時、共有メモリ４１にアクセス可能に構成されている。例えば、モータ２２２を含む所定の装置とロボット２１０とを同期制御する際に、当該所定の装置の動きに対応させてロボット２１０の動きを決定する必要がある場合には、モーション処理部３２による演算処理において当該所定の装置に関連する変数であり、共有メモリ４１に保持されている変数データを使用する必要がある。このような場合には、制御アプリケーション処理部３０は、ＩＥＣプログラム処理部４０を介して共有メモリ４１にアクセスし、当該変数データを取得する。

【００３１】

制御アプリケーション処理部３０による共有メモリ４１内の変数データの具体的な態様について説明する。第１の形態では、ＩＥＣプログラム処理部４０による指令値演算で使用される変数名と、制御アプリケーション処理部３０による指令値演算で使用される変数名とが同一に設定される。この場合、制御アプリケーション処理部３０においてモーション処理部３２による演算処理が行われる際に、アプリケーションプログラム５２に含まれる変数と同一名の変数の値を返すように、制御アプリケーション処理部３０からＩＥＣプログラム処理部４０に対して問合せ処理が行われる。問合せられたＩＥＣプログラム処理部４０は、同一名の変数が共有メモリ４１内に保持されている場合には、当該変数の変数データを制御アプリケーション処理部３０に対して返答する。この結果、モーション処理

部３２による演算処理が継続される。なお、同一名の変数が共有メモリ４１内に保持されていない場合には、ＩＥＣプログラム処理部４０は、エラー信号を制御アプリケーション処理部３０に対して返答する。

【００３２】

また、第２の形態では、ＩＥＣプログラム処理部４０による指令値演算で使用される変数名と、制御アプリケーション処理部３０による指令値演算で使用される変数名とは、異なる変数名で設定される。その上で、ＩＥＣプログラム処理部４０側の変数名と制御アプリケーション処理部３０側の変数名との相関を画定するテーブル（図３を参照）が、制御アプリケーション処理部３０に保持されている。この場合、制御アプリケーション処理部３０においてモーション処理部３２による演算処理が行われる際に、アプリケーションプログラム５２に含まれる変数（図３における「制御アプリケーション処理部側の変数」）に対応する、共有メモリ４１に保持されている変数（図３における「ＩＥＣプログラム処理部側の変数」）を特定し、その特定された変数の値を返すように、制御アプリケーション処理部３０からＩＥＣプログラム処理部４０に対して問合せ処理が行われる。問合せ後の返答については、第１の形態と同様である。

【００３３】

このように共有メモリ４１を介して、ＩＥＣプログラム処理部４０と制御アプリケーション処理部３０との間で、指令値演算のための変数を共有する形態を採用することで、ユーザは、所定の装置とロボット２１０とを同期制御するにあたって、ＩＥＣプログラム処理部４０と制御アプリケーション処理部３０との間のデータのやり取りを制御するプログラムを別途準備する必要は無くなり、以て、ユーザの負荷軽減が図られる。

【００３４】

＜プログラムの同期実行＞

制御装置１００においては、ＩＥＣプログラム５１とアプリケーションプログラム５２の同期実行が実現される。制御アプリケーション処理部３０のインタプリタ３４は、制御周期より長い周期、例えば、制御周期の２倍分の周期毎にアプリケーションプログラム５２を逐次実行する。ただし、ＩＥＣプログラム処理部４０のモーション処理部４２および制御アプリケーション処理部３０のモーション処理部３２は、いずれも同一の制御周期毎に指令値を演算する。したがって、制御装置１００からの指令値の出力は、いずれも予め定められた制御周期で同期して行われる。このように、ＩＥＣプログラム処理部４０および制御アプリケーション処理部３０は、アクチュエータの動きを連続的に制御するためのモーション処理部をそれぞれ有しており、これらのモーション処理部が同期して指令値を演算することで、ＩＥＣプログラム５１に従う制御およびアプリケーションプログラム５２に従う制御の両方を制御周期と同期させて実行することができ、これによって制御周期単位での精密な制御が実現される。

【００３５】

次に、制御装置１００におけるＩＥＣプログラム５１およびアプリケーションプログラム５２の実行タイミングの詳細について、図４に基づいて説明する。図４は、制御装置１００におけるプログラムの実行タイミングの一例を示す図である。なお、制御装置１００においては、プロセッサのリソースを考慮して、優先度が高い高優先度タスク（図４中の上段の処理）と優先度が低い低優先度タスク（図４中の下段の処理）が設定されている。具体的には、下位ネットワークインターフェイス６０、ＩＥＣプログラム処理部４０とそのモーション処理部４２の実行、および制御アプリケーション処理部３０のモーション処理部３２の実行は、高優先度タスクとして設定され、制御アプリケーション処理部３０のインタプリタ３４の実行は低優先度タスクとして設定されている。

【００３６】

すなわち、下位ネットワークインターフェイス６０に関連する入出力リフレッシュ処理Ｂ６０、ＩＥＣプログラム５１の実行処理Ｂ４０、ＩＥＣプログラム５１に従ってモーション処理部４２によって行われる指令値の演算処理Ｂ４２、アプリケーションプログラム５２に従ってモーション処理部３２によって行われる指令値の演算処理Ｂ３２は、高優先

度タスクとして実行される。一方、アプリケーションプログラム５２を逐次解釈する処理Ｂ３４は、低優先度タスクとして実行される。

【００３７】

ここで、高優先度タスクは、予め定められた制御周期Ｔ１毎に繰返し実行される。低優先度タスクは、各制御周期内で高優先度タスクが実行されていない期間に都度実行される。すなわち、制御周期毎に、高優先度タスクの実行時間が割当てられ、高優先度タスクの実行時間以外の時間において、低優先度タスクが実行される。

【００３８】

まず、高優先度タスクについて説明すると、各制御周期が到来すると、入出力リフレッシュ処理Ｂ６０が実行された後、ＩＥＣプログラム処理部４０によりＩＥＣプログラム５１の全体が実行（スキャン）されて、シーケンス制御についての１または複数の指令値が演算される（実行処理Ｂ４０）。併せて、モーション処理部４２によりＩＥＣプログラム５１に含まれるモーション命令に関するモーション処理が実行されて、モーション命令についての１または複数の指令値が演算される（実行処理Ｂ４２）。さらに、制御アプリケーション処理部３０のモーション処理部３２により、バッファ３３に格納されている中間コードに従ってロボット２１０の制御用のモーション指令が準備される（実行処理Ｂ３２）。以下、同様の処理が制御周期毎に繰返される。なお、モーション処理部３２がバッファ３３から中間コードを読み出すタイミングは、各制御周期でなくともよい。これは、読み出された中間コードは、複数の制御周期Ｔ１にわたって指令値を演算できるだけの命令を含む場合、その複数の制御周期Ｔ１においては中間コードの読み出しを一度で行える。

【００３９】

このように、ある制御周期における高優先度タスクの実行が完了すると、モータ２２２等に関するシーケンス制御についての指令値及びそのモーション制御についての指令値と、ロボット２１０に関する制御アプリケーションについての指令値のセットが用意される。これらの指令値は、基本的には、次の制御周期が到来すると、フィールド側に反映される。すなわち、ＩＥＣプログラム処理部４０および制御アプリケーション処理部３０は、同一の制御周期で入力データに応じた指令値を演算するので、入力に同期した出力を実現できる。

【００４０】

一方、低優先度タスクについては、制御アプリケーション処理部３０のインタプリタ３４は、アプリケーションプログラム５２を逐次実行する。すなわち、インタプリタ３４は、アプリケーションプログラム５２の読込みおよび解析を低優先度で実行する。インタプリタ３４がアプリケーションプログラム５２を解析処理して生成された中間コードは、バッファ３３の容量を考慮しながら、逐次、バッファ３３に格納される。バッファ３３に格納された中間コードは、制御アプリケーション処理部３０のモーション処理部３２により順次参照されて、演算処理Ｂ３２での指令値の生成に用いられる。このとき、インタプリタ３４は、高優先度タスクの演算周期である制御周期の整数倍分の中間コードを事前に余分に生成しておくことで、モーション処理部３２による処理に影響を与えることなく、制御アプリケーションに対する指令値を制御周期毎に演算できる。

【００４１】

また、インタプリタ３４は、予め定められた制御アプリケーション同期周期（制御周期の整数倍）が到来する前に、アプリケーションプログラム５２の解釈を一時停止する。その一時停止したタイミングで、ＩＥＣプログラム処理部４０と制御アプリケーション処理部３０との間でデータ同期を行うことで、双方に整合性をもつデータを共有する。このように、インタプリタ３４は、同期周期毎に、ＩＥＣプログラム処理部４０との間で共有するデータを更新する。共有データの更新に併せて、フィールド側から取得される入力データおよび出力データについても更新（データ同期）するようにしてもよい。これにより、制御アプリケーション処理部３０側でも、ＩＥＣプログラム処理部４０で取得されたデータを利用してロボット２１０の制御が可能となる。制御アプリケーション同期周期は、制御周期の整数倍に設定されれば、どのような長さであってもよい。制御アプリケーション

において要求される制御の精度などに応じて、適宜設定される。

【００４２】

次に、図５、図６に基づいて、制御装置１００によるフィールド機器（ロボット２１０及びモータ２２２）の処理の流れについて説明する。図５は、上述した高優先度タスクの処理に関するフローチャートであり、図６は、上述した低優先度タスクの処理に関するフローチャートである。

【００４３】

先ず、高優先度タスクの処理の流れについて説明する。制御周期Ｔ１が到来すると、下位ネットワークインターフェイス６０が入出力リフレッシュ処理を実行する（Ｓ１０１の処理）。これによって、直前の制御周期Ｔ１において演算された指令値（Ｂ４０、Ｂ４２、Ｂ３２等による指令値）がフィールド機器のアクチュエータ等へ出力されるとともに、フィールド機器からの入力データが取得される。続いて、Ｓ１０２において、今回の制御周期がデータ同期のタイミングと一致するか否かが判定される。そこで肯定判定されると、ＩＥＣプログラム処理部４０と制御アプリケーション処理部３０との間でデータ同期が実行される（Ｓ１０３の処理）。またＳ１０２で否定判定されると、処理はＳ１０４へ進む。

【００４４】

次に、Ｓ１０４では上記の実行処理Ｂ４０が行われ、続いてＳ１０５では上記の実行処理Ｂ４２が行われる。その後、Ｓ１０６で、ロボット２１０を制御するためのモーション指令の準備を行う、上記のロボット実行処理Ｂ３２が行われる。そして、ロボット実行処理Ｂ３２が行われる際に、アプリケーションプログラム５２に所定の変数が含まれている場合には、上述した第１の形態又は第２の形態に従って、共有メモリ４１内に保持されている、対応する変数の値が利用される。

【００４５】

図５に示す高優先度タスクの一連の処理により演算、準備された指令値は、次の制御周期Ｔ１が到来したときにフィールドへ出力される。そして、当該一連の処理の終了後であって次の制御周期Ｔ１が到来するまでの期間において、図６に示す低優先度タスクが実行されることになる。

【００４６】

図６に基づいて、低優先度タスクの処理の流れについて説明する。低優先度タスクは、インタプリタ３４によるアプリケーションプログラム５２の解釈処理に関するものである。先ず、Ｓ２０１で、制御アプリケーション処理部３０により、バッファ３３に中間コードが残っているか否かが判定される。当該判定を行うのは、バッファ３３の容量を超えた中間コードの生成を行わないようにするためである。Ｓ２０１で肯定判定されれば低優先度タスクは終了し、否定判定されれば処理はＳ２０２へ進む。Ｓ２０２では、インタプリタ３４によって、アプリケーションプログラム５２の一部が読み込まれる。例えば、アプリケーションプログラム５２を構成するコードの一行分が読み込まれる。そして、Ｓ２０３では、インタプリタ３４によって読込んだコードが解釈されて中間コードが生成される。生成された中間コードは、Ｓ２０４でバッファ３３に格納される。なお、Ｓ２０２～Ｓ２０４の処理に関し、実行対象となるアプリケーションプログラムが存在しない場合にはこれらの処理は行われず、結果として、バッファ３３には中間コードは格納されないことになる。このような一連の処理を有する低優先度タスクは、自身に対してプログラムの実行時間が割当てられている期間において繰返される。

【００４７】

このように図５、図６に示す一連の処理が行われることで、制御装置１００において、図４に示す実行タイミングで、ＩＥＣプログラム５１およびアプリケーションプログラム５２が実行されることになる。

【００４８】

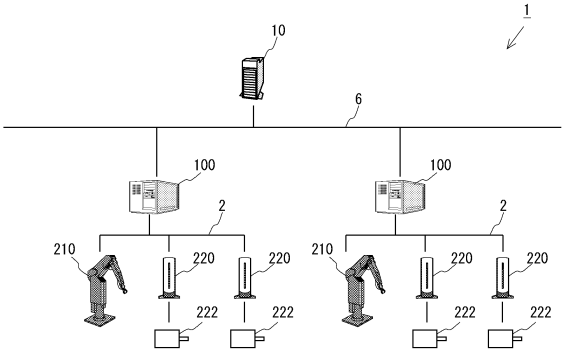
＜付記１＞

実行毎に全体が実行される第１の実行形式に従ったプログラム（５１）と、逐次実行さ

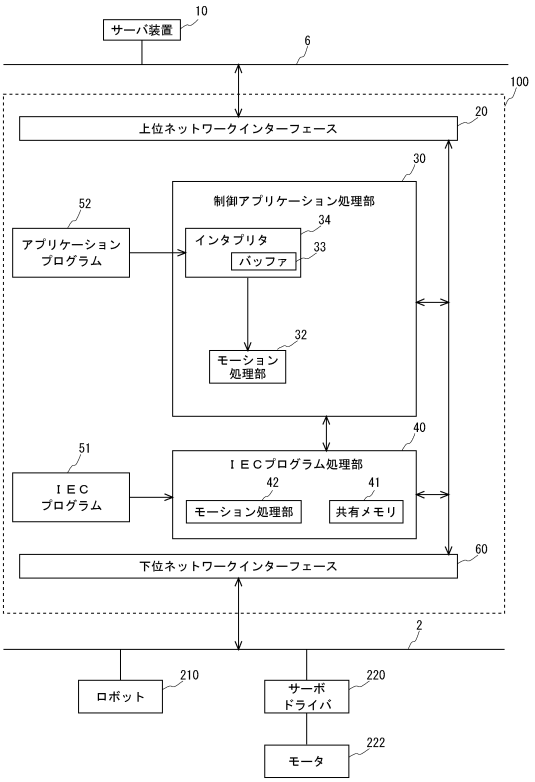
れる第２の実行形式に従ったプログラム（５２）と、を有する制御装置（１００）であって、
予め定められた制御周期毎に前記第１の実行形式に従ったプログラム（５１）を実行して第１の制御対象（２２２）を制御するための第１の指令値を演算可能に構成された第１処理部（４０）と、
前記第２の実行形式に従ったプログラム（５２）を実行してインタプリタ（３４）により該プログラムの少なくとも一部が解釈されて生成される中間コードに従って前記制御周期毎に第２の制御対象（２１０）を制御するための第２の指令値を演算可能に構成された第２処理部（３０）と、
前記第１の指令値および前記第２の指令値を前記制御周期毎に出力する出力部（６０）と、
を備え、
前記第１処理部（４０）と前記第２処理部（３０）のうちいずれか一方の処理部が、該第１処理部（４０）による演算と該第２処理部（３０）による演算の両方において使用可能な所定変数を保持し、該第１処理部（４０）と該第２処理部（３０）のうち他方の処理部は、該一方の処理部を介して該所定変数にアクセス可能に構成される、
制御装置。
【符号の説明】
【００４９】
２：フィールドネットワーク
６：上位ネットワーク
１０：サーバ装置
３０：制御アプリケーション処理部
４０：ＩＥＣプログラム処理部
５１：ＩＥＣプログラム
５２：アプリケーションプログラム
１００：制御装置
２１０：ロボット
２２０：サーボドライバ
２２２：モータ

【図面】

【図１】



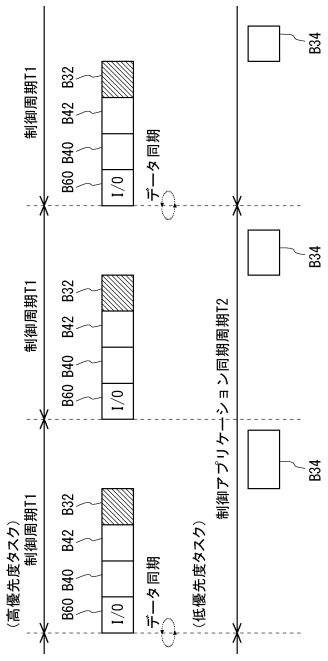
【図２】



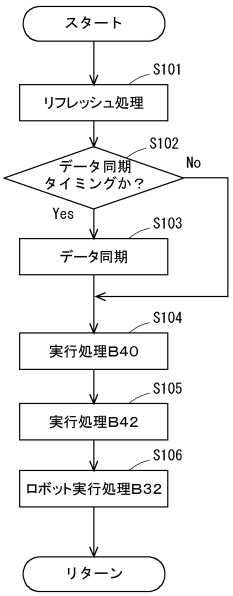
【図３】

制御アプリケーション処理部側の変数	IECプログラム処理部側の変数
rob_var1	plc_var1
rob_var2	plc_var2
rob_var3	plc_var3

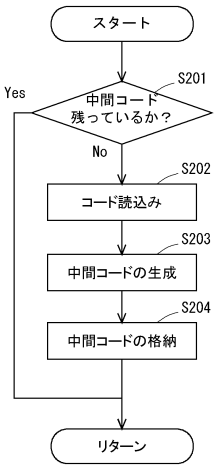
【図４】



【図５】


スタート
S101
リフレッシュ処理
S102
データ同期
タイミングか？
No
Yes
S103
データ同期
S104
実行処理B40
S105
実行処理B42
S106
ロボット実行処理B32
リターン

【図６】


スタート
S201
中間コード
残っているか？
Yes
No
S202
コード読込み
S203
中間コードの生成
S204
中間コードの格納
リターン

フロントページの続き

審査官　大古 健一
(56)参考文献　特開２０１９－３６０４３（ＪＰ，Ａ）
特開２０１８－１８５７７２（ＪＰ，Ａ）
特開平１１－２８８３０４（ＪＰ，Ａ）
特開平７－７２９２０（ＪＰ，Ａ）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０５Ｂ　１９／０４　－１９／０５