(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6806019号
(P6806019)
(45)発行日　令和2年12月23日(2020.12.23)　(24)登録日　令和2年12月8日(2020.12.8)

(51)Int.Cl.　　F I
G06F 9/48 (2006.01)　　G06F 9/48 3OOF
G06F 9/50 (2006.01)　　G06F 9/50 12OB

請求項の数 8　(全 36 頁)

(21)出願番号　特願2017-185146(P2017-185146)
(22)出願日　平成29年9月26日(2017.9.26)
(65)公開番号　特開2019-61466(P2019-61466A)
(43)公開日　平成31年4月18日(2019.4.18)
審査請求日　令和1年5月8日(2019.5.8)

(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　110001195
特許業務法人深見特許事務所
(72)発明者　島村　純児
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　若年　哲司
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　島村　知行
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】
１または複数のプロセッサを有する制御装置であって、
実行毎に全体がスキャンされる第１のプログラムと、逐次実行される第２のプログラムとを格納する記憶部と、
第１の周期毎に前記第１のプログラムを実行して第１の制御指令を出力するプログラム実行部と、
前記第１の周期より長い第２の周期毎に前記第２のプログラムの少なくとも一部を構文解析して内部コマンドを逐次生成する解析部と、
前記解析部により生成された内部コマンドに従って前記第１の周期毎に第２の制御指令を出力する指令演算部と、
予め設定された優先度に基づいて、１または複数のタスクに対してプロセッサリソースを割当てるスケジューラとを備え、
前記スケジューラには、少なくとも、前記プログラム実行部および前記指令演算部による処理実行を含む第１の優先度を有する第１のタスクと、前記解析部による処理実行を含む前記第１の優先度より低い第２の優先度を有する第２のタスクと、前記第１のタスクおよび前記第２のタスクとは異なる処理内容の実行を含む第３の優先度を有する第３のタスクとが設定されており、
前記解析部における処理状況を監視するとともに、前記解析部における処理状況が予め定められた条件を満たすと、前記第２のタスクに設定されている前記第２の優先度と前記

第３のタスクに設定されている前記第３の優先度との間で、優先順位を当該条件に応じて変更する優先度変更部とを備え、
前記優先度変更部は、前記解析部による内部コマンドの生成処理に係る負荷を示す情報に基づいて、前記解析部における処理状況を判断し、
前記負荷を示す情報は、前記解析部による前記第２のプログラムの構文解析に要する時間を含む、制御装置。
【請求項２】
１または複数のプロセッサを有する制御装置であって、
実行毎に全体がスキャンされる第１のプログラムと、逐次実行される第２のプログラムとを格納する記憶部と、
第１の周期毎に前記第１のプログラムを実行して第１の制御指令を出力するプログラム実行部と、
前記第１の周期より長い第２の周期毎に前記第２のプログラムの少なくとも一部を構文解析して内部コマンドを逐次生成する解析部と、
前記解析部により生成された内部コマンドに従って前記第１の周期毎に第２の制御指令を出力する指令演算部と、
予め設定された優先度に基づいて、１または複数のタスクに対してプロセッサリソースを割当てるスケジューラとを備え、
前記スケジューラには、少なくとも、前記プログラム実行部および前記指令演算部による処理実行を含む第１の優先度を有する第１のタスクと、前記解析部による処理実行を含む前記第１の優先度より低い第２の優先度を有する第２のタスクと、前記第１のタスクおよび前記第２のタスクとは異なる処理内容の実行を含む第３の優先度を有する第３のタスクとが設定されており、
前記解析部における処理状況を監視するとともに、前記解析部における処理状況が予め定められた条件を満たすと、前記第２のタスクに設定されている前記第２の優先度と前記第３のタスクに設定されている前記第３の優先度との間で、優先順位を当該条件に応じて変更する優先度変更部とを備え、
前記優先度変更部は、前記解析部による内部コマンドの生成処理に係る負荷を示す情報に基づいて、前記解析部における処理状況を判断し、
前記負荷を示す情報は、前記解析部による前記第２のプログラムの構文解析に要する時間が前記第２の周期の長さを超過したか否かの情報を含む、制御装置。
【請求項３】
１または複数のプロセッサを有する制御装置であって、
実行毎に全体がスキャンされる第１のプログラムと、逐次実行される第２のプログラムとを格納する記憶部と、
第１の周期毎に前記第１のプログラムを実行して第１の制御指令を出力するプログラム実行部と、
前記第１の周期より長い第２の周期毎に前記第２のプログラムの少なくとも一部を構文解析して内部コマンドを逐次生成する解析部と、
前記解析部により生成された内部コマンドに従って前記第１の周期毎に第２の制御指令を出力する指令演算部と、
予め設定された優先度に基づいて、１または複数のタスクに対してプロセッサリソースを割当てるスケジューラとを備え、
前記スケジューラには、少なくとも、前記プログラム実行部および前記指令演算部による処理実行を含む第１の優先度を有する第１のタスクと、前記解析部による処理実行を含む前記第１の優先度より低い第２の優先度を有する第２のタスクと、前記第１のタスクおよび前記第２のタスクとは異なる処理内容の実行を含む第３の優先度を有する第３のタスクとが設定されており、
前記解析部における処理状況を監視するとともに、前記解析部における処理状況が予め定められた条件を満たすと、前記第２のタスクに設定されている前記第２の優先度と前記

第３のタスクに設定されている前記第３の優先度との間で、優先順位を当該条件に応じて変更する優先度変更部とを備え、
前記優先度変更部は、前記解析部による内部コマンドの生成処理に係る負荷を示す情報に基づいて、前記解析部における処理状況を判断し、
前記解析部は、前記第２のプログラムを構文解析して軌跡上の通過点を算出し、算出した通過点に基づいて内部コマンドを生成するように構成されており、
前記負荷を示す情報は、前記解析部が前記内部コマンドを生成するために事前に算出した通過点の数を含む、制御装置。
【請求項４】
１または複数のプロセッサを有する制御装置であって、
実行毎に全体がスキャンされる第１のプログラムと、逐次実行される第２のプログラムとを格納する記憶部と、
第１の周期毎に前記第１のプログラムを実行して第１の制御指令を出力するプログラム実行部と、
前記第１の周期より長い第２の周期毎に前記第２のプログラムの少なくとも一部を構文解析して内部コマンドを逐次生成する解析部と、
前記解析部により生成された内部コマンドに従って前記第１の周期毎に第２の制御指令を出力する指令演算部と、
予め設定された優先度に基づいて、１または複数のタスクに対してプロセッサリソースを割当てるスケジューラとを備え、
前記スケジューラには、少なくとも、前記プログラム実行部および前記指令演算部による処理実行を含む第１の優先度を有する第１のタスクと、前記解析部による処理実行を含む前記第１の優先度より低い第２の優先度を有する第２のタスクと、前記第１のタスクおよび前記第２のタスクとは異なる処理内容の実行を含む第３の優先度を有する第３のタスクとが設定されており、
前記解析部における処理状況を監視するとともに、前記解析部における処理状況が予め定められた条件を満たすと、前記第２のタスクに設定されている前記第２の優先度と前記第３のタスクに設定されている前記第３の優先度との間で、優先順位を当該条件に応じて変更する優先度変更部とを備え、
前記優先度変更部は、前記解析部による内部コマンドの生成処理に係る負荷を示す情報に基づいて、前記解析部における処理状況を判断し、
前記負荷を示す情報は、前記第２のプログラムに記述される特殊コマンドを含む、制御装置。
【請求項５】
前記優先度変更部は、
前記解析部による内部コマンドの生成処理の余裕がなくなった場合に、前記第２のタスクに設定されている前記第２の優先度を、前記第３のタスクに設定されている前記第３の優先度よりも高め、
前記解析部による内部コマンドの生成処理の余裕が生じた場合に、前記第２のタスクに設定されている前記第２の優先度を、前記第３のタスクに設定されている前記第３の優先度よりも低くする、請求項１～４のいずれか１項に記載の制御装置。
【請求項６】
前記負荷を示す情報は、前記解析部により生成された内部コマンドのうち、前記指令演算部により未だ処理されていない内部コマンドの数を含む、請求項１～５のいずれか１項に記載の制御装置。
【請求項７】
前記優先度変更部は、前記第２のタスクに設定されている前記第２の優先度を上げた後に、予め定められた別の条件が満たされると、前記第２の優先度を元に戻す、請求項１～６のいずれか１項に記載の制御装置。
【請求項８】

前記解析部は、複数の前記第２のプログラムのそれぞれについて構文解析を実行してそれぞれの内部コマンドを逐次生成するように構成されており、
前記優先度変更部は、前記複数の第２のプログラムのそれぞれの構文解析に関連付けられた複数の前記第２のタスクの間で、優先度を変更する、請求項１～７のいずれか１項に記載の制御装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、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】
生産現場においては、様々なＦＡ（Factory Automation）技術が広く普及している。このようなＦＡシステムは、ＰＬＣ（プログラマブルコントローラ）といった制御装置だけではなく、ＣＮＣ（Computer Numerical Control：コンピュータ数値制御）およびロボットなどの高度な動作が可能な各種アクチュエータなどを含む。このようなＦＡシステムはますます大規模化する傾向にある。
【０００３】
例えば、特開２００１－０２７９０４号公報（特許文献１）は、複数のＣＮＣ装置を連結させることで、制御する軸数を増大させ、かつ、複数のＣＮＣ装置間の同期を取ることができる数値制御システムを開示する。この数値制御システムにおいて、マスタユニットおよび１または複数のスレーブユニットがケーブルを介して接続されている。マスタユニットおよびスレーブユニットの各々は、ラダープログラムを実行するＰＭＣ用プロセッサおよびＣＮＣ用プロセッサを有している。
【０００４】
また、特開２０１６－０９２５４４号公報（特許文献２）は、マスタ制御装置により第１制御対象物を制御するとともに、当該マスタ制御装置とネットワークを介して接続されるスレーブ制御装置により第１制御対象物と異なる第２制御対象物を制御する制御システムを開示する。この制御システムは、マスタ制御装置およびスレーブ制御装置を含む。マスタ制御装置およびスレーブ制御装置の各々は、演算処理部を有するコントローラとは別に、別の演算処理部を有するロボットコントローラを含んでいる。
【０００５】
近年のＩＣＴ（Information and Communication Technology）の進歩に伴って、制御装置の処理能力も飛躍的に向上しつつある。従来技術においては複数の専用装置を用いて実現されていたＦＡシステムをより少ない数の制御装置で実現したいというニーズも生じている。
【先行技術文献】
【特許文献】
【０００６】
【特許文献１】特開２００１－０２７９０４号公報
【特許文献２】特開２０１６－０９２５４４号公報
【発明の概要】
【発明が解決しようとする課題】
【０００７】
上述した特許文献１および特許文献２に開示される構成においては、複数のプロセッサまたは演算処理部が分散配置された構成が採用されており、それぞれのプロセッサまたは演算処理部の間で同期処理を実現するために構成が冗長になり得る。
【０００８】
一方で、実行形式の異なる複数種類のプログラム（例えば、ラダープログラム、ＣＮＣプログラム、ロボットプログラムなど）を単一の制御装置で実現しようとすると、限られた処理リソースを効率的に利用する必要がある。本発明は、このようなニーズに対応でき

る制御装置を提供することを一つの目的としている。
【課題を解決するための手段】
【０００９】
本開示の一例によれば、１または複数のプロセッサを有する制御装置が提供される。制御装置は、実行毎に全体がスキャンされる第１のプログラムと、逐次実行される第２のプログラムとを格納する記憶部と、第１の周期毎に第１のプログラムを実行して第１の制御指令を出力するプログラム実行部と、第１の周期より長い第２の周期毎に第２のプログラムの少なくとも一部を構文解析して内部コマンドを逐次生成する解析部と、解析部により生成された内部コマンドに従って第１の周期毎に第２の制御指令を出力する指令演算部と、予め設定された優先度に基づいて、１または複数のタスクに対してプロセッサリソースを割当てるスケジューラとを含む。スケジューラには、少なくとも、プログラム実行部および指令演算部による処理実行を含む第１の優先度を有する第１のタスクと、解析部による処理実行を含む第１の優先度より低い第２の優先度を有する第２のタスクと、第１のタスクおよび第２のタスクとは異なる処理内容の実行を含む第３の優先度を有する第３のタスクとが設定されている。制御装置は、さらに、解析部における処理状況を監視するとともに、解析部における処理状況が予め定められた条件を満たすと、第２のタスクに設定されている第２の優先度を当該条件に応じて変更する優先度変更部を含む。
【００１０】
この開示によれば、複数のタスクが共通のプロセッサリソースを共有する場合に、制御指令を生成するためのソースとなる内部コマンドの生成が間に合わないといった事態を防止して、第１のプログラムおよび第２のプログラムに従う制御指令の第１の周期毎の出力を保障できる。
【００１１】
上述の開示において、優先度変更部は、解析部による内部コマンドの生成処理に係る負荷を示す情報に基づいて、解析部における処理状況を判断してもよい。
【００１２】
この開示によれば、実質的に優先度の変更対象となる解析部の負荷状況を評価することができるので、優先度変更の処理の安定性を高めることができる。
【００１３】
上述の開示において、負荷を示す情報は、解析部による第２のプログラムの構文解析に要する時間を含んでいてもよい。
【００１４】
この開示によれば、解析部による第２のプログラムの構文解析の負荷状況を客観的あるは定量的に評価できるので、解析部の負荷が過大になってしまう前に優先度を高めて、内部コマンドの生成が間に合わないといった事態の発生を予防できる。
【００１５】
上述の開示において、負荷を示す情報は、解析部による第２のプログラムの構文解析に要する時間が第２の周期の長さを超過したか否かの情報を含んでいてもよい。
【００１６】
この開示によれば、解析部による第２のプログラムの構文解析が本来の周期を超えているという事実に基づいて優先度が変更されるので、本来的に必要な状況にはないときに、優先度を変更してしまうといった事態を防止できる。
【００１７】
上述の開示において、解析部は、第２のプログラムを構文解析して軌跡上の通過点を算出し、算出した通過点に基づいて内部コマンドを生成するように構成されていてもよい。負荷を示す情報は、解析部が内部コマンドを生成するために事前に算出した通過点の数を含んでいてもよい。
【００１８】
この開示によれば、解析部による第２のプログラムを構文解析自体の処理負荷を評価することができ、内部コマンドの生成が間に合わないといった事態を防止できる。

【００１９】
上述の開示において、負荷を示す情報は、解析部により生成された内部コマンドのうち、指令演算部により未だ処理されていない内部コマンドの数を含んでいてもよい。
【００２０】
この開示によれば、内部コマンドの生成が追い付かずに、解析部による制御指令の出力が中断してしまうような事態を防止できる。
【００２１】
上述の開示において、負荷を示す情報は、第２のプログラムに記述される特殊コマンドを含んでいてもよい。
【００２２】
この開示によれば、第２のプログラムに記述されるコマンドのうち特定のコマンドについて、優先度を高めて処理することを直接的に命令できるので、第２のプログラムの構文解析および内部コマンドの生成をより確実に実現できる。
【００２３】
上述の開示において、優先度変更部は、第２のタスクに設定されている第２の優先度を上げた後に、予め定められた別の条件が満たされると、第２の優先度を元に戻すようにしてもよい。
【００２４】
この開示によれば、第２のプログラムを構文解析するにあたって、必要なときだけ、解析部に対してより多くのプロセッサリソースを割当てる一方で、必要ないときには、解析部がプロセッサリソースを占有することなく、他のタスクに対してプロセッサリソースを割当てることができる。
【００２５】
上述の開示において、解析部は、複数の第２のプログラムのそれぞれについて構文解析を実行してそれぞれの内部コマンドを逐次生成するように構成されていてもよい。優先度変更部は、複数の第２のプログラムのそれぞれの構文解析に関連付けられた複数の第２のタスクの間で、優先度を変更してもよい。
【００２６】
この開示によれば、制御装置において、複数の第２のプログラムに従う制御を並列的に実行するにあたって、第２のプログラム間で優先度を調停できる。
【発明の効果】
【００２７】
本発明によれば、単一の制御装置上で実行形式の異なる複数種類のプログラムを効率的に動作させることができる。
【図面の簡単な説明】
【００２８】
【図１】本実施の形態に係る制御装置における処理の概要を説明するためのタイムチャートである。
【図２】本実施の形態に係る制御システムの全体構成例を示す模式図である。
【図３】本実施の形態に係る制御装置のハードウェア構成例を示すブロック図である。
【図４】本実施の形態に係る制御装置の機能構成例を示すブロック図である。
【図５】本実施の形態に係る制御装置における内部コマンドの生成処理を説明するための模式図である。
【図６】本実施の形態に係る制御装置におけるタスクの実行タイミングの一例を示すタイムチャートである。
【図７】本実施の形態に係る制御装置における低優先タスクのタスク実行周期超過の一例を示すタイムチャートである。
【図８】本実施の形態に係る制御装置における処理実行に要する時間を示す変数リストの一例を示す図である。
【図９】本実施の形態に係る制御装置における内部コマンドの生成処理の一例を説明する

ための模式図である。
【図１０】本実施の形態に係る制御装置における内部コマンドを生成するための通過点の処理例を示す模式図である。
【図１１】本実施の形態に係る制御装置における内部コマンドから制御指令を演算する処理例を示す模式図である。
【図１２】本実施の形態に係る制御装置において処理されるアプリケーションプログラムの一例を示す模式図である。
【図１３】本実施の形態に係る制御装置における優先度変更の処理手順を示すフローチャートである。
【図１４】本実施の形態に係る制御装置における複数のアプリケーションプログラムを並列的に実行させる場合の実行タイミングの一例を示すタイムチャートである。
【図１５】本実施の形態に係る制御装置が提供するオーバーラップ動作を説明するための模式図である。
【図１６】本実施の形態に係る制御装置におけるオーバーラップ動作を実現するためのプログラム例を示す模式図である。
【図１７】時間をオーバーラップ条件とした場合のオーバーラップ動作の一例を示す模式図である。
【図１８】位置をオーバーラップ条件とした場合のオーバーラップ動作の一例を示す模式図である。
【図１９】本実施の形態に係る制御装置においてオーバーラップ動作を実現するための機能構成を示す模式図である。
【発明を実施するための形態】
【００２９】
本発明の実施の形態について、図面を参照しながら詳細に説明する。なお、図中の同一または相当部分については、同一符号を付してその説明は繰返さない。
【００３０】
＜Ａ．適用例＞
まず、図１を参照して、本発明が適用される場面の一例について説明する。図１は、本実施の形態に係る制御装置１００における処理の概要を説明するためのタイムチャートである。
【００３１】
図１を参照して、本実施の形態に係る制御装置１００においては、少なくとも、実行毎に全体がスキャンされる第１のプログラムと、逐次実行される第２のプログラムとのいずれもが実行される。
【００３２】
本実施の形態においては、第１のプログラムの一例として、「ＩＥＣプログラム」を想定する。制御装置１００は、予め定められた制御周期Ｔ１（第１の周期）毎にＩＥＣプログラムを実行して制御指令を出力する。
【００３３】
本明細書において、ＩＥＣプログラムは、実行毎に全体がスキャンされて、実行毎に制御指令を出力できるプログラムを包含する概念である。ＩＥＣプログラムは、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プログラムには、シーケンス命令および／またはモーション命令を含み得る。なお、ＩＥＣプログラムとしては、国際規格ＩＥＣ６１１３１－３に従って記述された命令に限らず、ＰＬＣ（プログラマブルコントローラ）の製造メーカまたはベンダーなどが独自に規定した命令を含むようにしてもよい。このように、ＩＥＣプログラムは、即時性および高速性が要求される制御に好適である。
【００３４】
ＩＥＣプログラムは、制御対象やアプリケーションに応じて、ユーザによって任意に作

成されるものであるため、以下の説明においては、便宜上、ＩＥＣプログラムを「ユーザプログラム」とも称す。

【００３５】

本明細書において、「シーケンス命令」は、入力値、出力値、内部値などを演算する１または複数の論理回路により記述される１または複数の命令を包含する用語である。１回の制御周期Ｔ１において、「シーケンス命令」は、先頭から最終まで実行され、次の制御周期Ｔ１において、「シーケンス命令」は、先頭から最終までが再度実行される。

【００３６】

本明細書において、「モーション命令」は、サーボモータなどのアクチュエータに対して、位置、速度、加速度、加加速度、角度、角速度、角加速度、角加加速度などの数値を制御指令として演算するための１または複数の命令を包含する用語である。「モーション命令」についても、１回の制御周期Ｔ１において、ファンクションブロックや数値演算式などにより記述されるモーション命令のプログラム（モーションプログラム）の先頭から最終までが実行される。

【００３７】

制御周期Ｔ１毎にユーザプログラムを実行することで出力される制御指令は、典型的には、シーケンス命令に従って決定されるデジタル出力のオン／オフ、ならびに、モーション命令に従って演算されるアナログ出力を含む。

【００３８】

本実施の形態においては、第２のプログラムの一例として、「アプリケーションプログラム」を想定する。

【００３９】

本明細書において、「アプリケーションプログラム」は、逐次実行される任意のプログラムを包含する。典型的には、「アプリケーションプログラム」は、１行ずつ逐次実行して実行するインタプリタ方式で実行可能な任意の言語で記述されたプログラムを包含する。以下の説明においては、このような「アプリケーションプログラム」の一例として、ＣＮＣ（Computer Numerical Control：コンピュータ数値制御）における挙動を記述するプログラム（以下、「ＮＣプログラム」とも称する。）がある。また、「アプリケーションプログラム」の別の一例として、ロボットを制御するためのプログラムがある。

【００４０】

ＮＣプログラムは、予め定められた言語に従って記述される。一例として、ＮＣプログラムは「Ｇ言語」を用いて記述され、ロボットを制御するためのプログラムは、専用のロボット言語を用いて記述されることが多い。

【００４１】

以下の説明において、例えば、ＣＮＣ（Computer Numerical Control：コンピュータ数値制御）および／またはおよびロボットを用いた、特定の加工または動作を行う装置または機械ならびにそれらの制御を包含して、「制御アプリケーション」と称することもある。

【００４２】

制御装置１００は、制御周期Ｔ１（第１の周期）より長いアプリケーション実行周期Ｔ２（第２の周期）毎にアプリケーションプログラムの少なくとも一部を構文解析して内部コマンドを逐次生成し、生成された内部コマンドに従って制御周期Ｔ１毎に制御指令を出力する。このように、制御装置１００は、アプリケーションプログラムを構文解析して内部コマンドを逐次生成し、その内部コマンドに基づいて、制御周期Ｔ１毎に制御指令を出力する。

【００４３】

逐次実行されるアプリケーションプログラムは、本来的に、一定周期毎に制御指令を出力する処理には不向きであるため、制御周期Ｔ１毎の制御指令の出力に適した内部コマンドが生成される。このような内部コマンドは、制御周期Ｔ１毎に制御指令を出力できるものであれば、どのようなコードであってもよい。典型的には、内部コマンドは、時間を入

力変数とする１または複数の関数、あるいは、時間を引数とする１または複数の命令であってもよい。

【００４４】

制御装置１００においては、ユーザプログラムに含まれるシーケンス命令に従って制御指令を出力する処理、ユーザプログラムに含まれるモーション命令に従って制御指令を出力する処理、アプリケーションプログラムを構文解析して内部コマンドを生成する処理、内部コマンドに従って制御指令を出力する処理が実行される。制御装置１００においては、これらの処理に加えて、関連する処理も実行される。すなわち、制御装置１００においては、共通の処理資源を用いて、複数の処理が実行される。

【００４５】

このような複数の処理の実行は、スケジューラと称される機能によって管理される。スケジューラは、制御装置１００が有している１または複数のプロセッサの処理時間（以下、「プロセッサリソース」とも称す。）を対象の処理に割当てる。

【００４６】

本実施の形態においては、１または複数の処理からなる「タスク」の単位でプロセッサリソースの割当てが管理される。各タスクには、優先度が予め設定されており、スケジューラは、予め設定された優先度に基づいて、１または複数のタスクにプロセッサリソースを割当てる。

【００４７】

各タスクには、実行周期を設定してもよい。実行周期が設定されている場合には、スケジューラは、予め設定された実行周期に従って、タスクに設定されている処理をトリガする。

【００４８】

図１（Ａ）には、優先度別に設定された５つのタスク（第１～第５タスク）の例を示す。

【００４９】

第１タスク１０は、フィールドとの間で入力値および制御指令を更新する入出力リフレッシュ処理（図１において「Ｏ／Ｉ」と記載する。）を含む。

【００５０】

第２タスク１８は、シーケンス命令実行処理１２と、アプリケーションプログラムに従って制御指令を出力する出力処理１４と、ユーザプログラムに含まれるモーション命令に従って制御指令を出力する出力処理１６とを含む。スケジューラには、相対的に高い優先度を有する第２タスク１８として、アプリケーションプログラム実行および制御指令出力の処理実行が設定されている。

【００５１】

第１タスク１０および第２タスク１８は、制御周期Ｔ１毎にトリガされる。そのため、第１タスク１０および第２タスク１８に対しては、相対的に高い優先度が設定されている。以下の説明においては、第１タスク１０および第２タスク１８を、「高優先タスク」とも称す。

【００５２】

第３タスク２０は、アプリケーションプログラムを構文解析して内部コマンドを逐次生成する処理を含む。すなわち、スケジューラには、相対的に低い優先度を有する第３タスク２０として、アプリケーションプログラムの構文解析の処理実行が設定されている。

【００５３】

第４タスク２２は、制御装置１００が提供する任意の処理を含む。すなわち、スケジューラには、相対的に低い優先度を有する第３タスク２０として、第２タスク１８および第３タスク２０とは異なる処理内容の実行が設定されている。

【００５４】

第５タスク２４は、制御装置１００と外部装置との間のデータ通信などの処理（以下、「システムサービス」とも称す。）を含む。

【００５５】
第３タスク２０、第４タスク２２、および第５タスク２４は、高優先タスクが実行されていない期間に実行される。以下の説明においては、第３タスク２０、第４タスク２２、および第５タスク２４を、「低優先タスク」とも称す。
【００５６】
第１～第５タスクについては、それぞれ優先度が設定されており、各設定された優先度に従って、プロセッサリソースが割当てられる。
【００５７】
図１（Ａ）には、第４タスク２２に対して、第３タスク２０より高い優先度が設定されている例を示す。図１（Ａ）に示す例においては、第１タスク１０および第２タスク１８が制御周期Ｔ１毎に繰返し実行される。各制御周期Ｔ１において、第１タスク１０および第２タスク１８が実行されていない期間において、第３～第５タスクが優先度に応じて実行される。
【００５８】
図１（Ａ）に示す例においては、第４タスク２２に対しては、制御周期Ｔ１毎にプロセッサリソースが割当てられている。一方、第３タスク２０に対しては、第１タスク１０、第２タスク１８、第４タスク２２のいずれもが実行されていない期間においてのみ、プロセッサリソースが割当てられるので、第３タスク２０の１回の処理完了には、３回の制御周期Ｔ１に相当する時間が必要になっている。すなわち、第３タスク２０がトリガされてから、指定された処理を完了するまでには、制御周期Ｔ１の３倍の時間が必要である。
【００５９】
上述したように、第３タスク２０として設定されている処理は、アプリケーションプログラムを構文解析して内部コマンドを逐次生成するものであり、この逐次生成される内部コマンドは、制御指令の出力に用いられる。
【００６０】
このように、第３タスク２０においては、アプリケーションプログラムを構文解析して内部コマンドが生成され、第２タスク１８においては、第３タスク２０において生成される内部コマンドが制御指令の演算に逐次使用される（すなわち、制御周期Ｔ１毎に「消費」される）。すなわち、第３タスク２０による内部コマンドの生成と、第２タスク１８による内部コマンドの使用との間で競争が生じる。
【００６１】
第３タスク２０に対して十分なプロセッサリソースが割当てられている場合においては、第２タスク１８による内部コマンドの使用に先だって、ある程度の量の内部コマンドが事前に生成できる。一方、第３タスク２０に対して十分なプロセッサリソースを割当てることができない場合においては、第２タスク１８による内部コマンドの使用が勝り、第３タスク２０による内部コマンドの生成を待つような事態も生じ得る。
【００６２】
そこで、本実施の形態に係る制御装置１００においては、第３タスク２０（アプリケーションプログラムに対する構文解析処理）における処理状況を監視するとともに、当該処理状況が予め定められた条件を満たすと、第３タスク２０に設定されている優先度を当該条件に応じて変更する。
【００６３】
例えば、内部コマンドの生成に余裕がなくなった場合などには、第３タスク２０に設定されている優先度を上げ、逆に、内部コマンドの生成に十分な余裕が生じた場合などには、第３タスク２０に設定されている優先度を下げるような処理が可能である。
【００６４】
図１（Ｂ）には、第３タスク２０の優先度を図１（Ａ）の場合の優先度より高く設定したときの処理例を示す。図１（Ｂ）においては、第３タスク２０の優先度は、第４タスク２２より高く設定されている。
【００６５】

図１（Ｂ）に示す例においては、第１タスク１０および第２タスク１８のいずれもが実行されていない期間において、第３タスク２０に対して、プロセッサリソースが優先的に割当てられる。その結果、図１（Ａ）の場合に比較して、より多くのプロセッサリソースが第３タスク２０に割当てられるようになり、第３タスク２０がトリガされてから制御周期Ｔ１の２倍の時間で指定された処理を完了できる。すなわち、アプリケーションプログラムの処理対象のコードから内部コマンドをより短い時間で生成できる。

【００６６】

図１（Ｂ）においては、図１（Ａ）の場合に比較して、より多くのプロセッサリソースを割当てることもでき、この場合には、１回の処理実行によって、より多くの内部コマンドの量を生成できる。

【００６７】

以上のように、アプリケーションプログラムを構文解析して内部コマンドを生成する処理を含む第３タスク２０の優先度を、その処理状況に応じて動的に変更することで、途中途切れることなく、制御周期Ｔ１毎の制御指令の出力を継続できる。

【００６８】

本実施の形態に係る制御装置１００によれば、例えば、ＣＮＣ工作機械やロボットなどをアプリケーションプログラムに従って制御するような場合において、ユーザプログラムに従う制御指令の更新周期と同期して、位置や速度などの制御指令を出力することができる。そのため、ＣＮＣ工作機械および／またはロボットと関連する搬送装置などとを同期させた制御を実現できる。このとき、上述したような構成を採用することで、ＣＮＣ工作機械および／またはロボットに対する制御指令を出力するための内部コマンドを、切れ目なく生成することができ、ＣＮＣ工作機械および／またはロボットが途中で一旦停止するような事態を生じさせることなく、正確かつ高速な制御および生産を実現できる。

【００６９】

図１においては、説明を簡素化するために、単一のプロセッサリソースを時分割して、複数のタスクに割当てる構成を説明したが、本発明の適用はこのような環境に限定されることはなく、公知のプログラム実行環境に適用できる。例えば、マルチコアまたはマルチプロセッサ環境においては、複数のプロセッサリソースをそれぞれ時分割して、必要なタスクに割当てることもできる。このような場合であっても、プロセッサリソースの割当ては、優先度に基づいて行われるため、上述したような優先度の変更処理を適用することによって、上述したのと同様の作用効果を奏することができる。

【００７０】

以下、本発明のより具体的な応用例として、本実施の形態に係る制御装置１００のより詳細な構成および処理について説明する。

【００７１】

＜Ｂ．制御システムの全体構成例＞

まず、本実施の形態に係る制御装置１００を含む制御システム１の全体構成例について説明する。図２は、本実施の形態に係る制御システム１の全体構成例を示す模式図である。図２には、本実施の形態に係る制御装置１００を中心とした制御システム１を示す。

【００７２】

図２を参照して、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り、典型的には、ＰＬＣ（プログラマブルコントローラ）として具現化されてもよい。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５００と接続されてもよい。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５００との間でデータを遣り取りする。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、以下の説明においては、「フィールドネットワーク」と総称する。すなわち、本明細書の「フィールドネットワーク」は、狭義の「フィールドネットワーク」に加えて「フィールドバス」を含み得る概念である。

【００７３】
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）、ＥｔｈｅｒＮｅｔ／ＩＰ（登録商標）、ＤｅｖｉｃｅＮｅｔ（登録商標）、ＣｏｍｐｏＮｅｔ（登録商標）などが知られている。
【００７４】
フィールドネットワーク２には、任意のフィールド機器５００を接続することができる。フィールド機器５００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遣り取りする入出力装置などを含む。
【００７５】
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５００との間でデータが遣り取りされることになるが、これらの遣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。このようなデータの遣り取りは、フィールド機器５００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。）を制御装置１００へ送信する処理、および、制御装置１００からフィールド機器５００に対する制御指令などのデータ（以下、「出力データ」とも称す。）を送信する処理を含む。このような遣り取りされるデータの更新処理が上述の「入出力リフレッシュ処理」に相当する。
【００７６】
図２に示す構成例においては、フィールド機器５００は、リモートＩ／Ｏ（Input/Output）装置５１０と、ロボット５２０およびロボットコントローラ５２２と、ＣＮＣ工作機械５３０と、サーボドライバ５４０およびサーボモータ５４２とを含む。
【００７７】
サーボモータ５４２は、コンベア５４４を駆動して、ＣＮＣ工作機械５３０の前に配置されたワークテーブル５４６にワークＷを搬送する。ロボット５２０は、ワークテーブル５４６上の処理前のワークＷをＣＮＣ工作機械５３０内に配置し、ＣＮＣ工作機械５３０での処理済ワークＷを取り出してワークテーブル５４６上に配置する。
【００７８】
フィールド機器５００としては、これらに限られることなく、入力データを収集する任意のデバイス（例えば、視覚センサなど）、ならびに、出力データに従う何らかの作用を与える任意のデバイス（例えば、インバータ装置など）などを採用することができる。
【００７９】
リモートＩ／Ｏ装置５１０は、典型的には、フィールドネットワーク２を介して通信を行う通信カプラと、入力データの取得および出力データの出力を行うための入出力部（以下、「Ｉ／Ｏユニット」とも称す。）とを含む。
【００８０】
リモートＩ／Ｏ装置５１０には、入力リレーや各種センサ（例えば、アナログセンサ、温度センサ、振動センサなど）などの入力データを収集する装置、および、出力リレー、コンタクタ、サーボドライバ、および、その他任意のアクチュエータなどのフィールドに対して何らかの作用を与える装置が接続される。
【００８１】
ロボットコントローラ５２２は、制御装置１００からの制御指令（位置指令または速度指令など）に従って、軌跡計算および各軸の角度計算などを行うとともに、計算結果に従って、ロボット５２０を構成するサーボモータなどを駆動する。
【００８２】
ＣＮＣ工作機械５３０は、位置や速度などを指定するプログラムに従って、マシニングセンタなどを制御することで、任意の対象物を加工する。ＣＮＣ工作機械５３０は、典型的には、旋盤加工、フライス盤、放電加工などの加工装置を含む。

【００８３】
サーボドライバ５４０は、制御装置１００からの制御指令（例えば、位置指令または速度指令など）に従って、サーボモータ５４２を駆動する。
【００８４】
制御装置１００は、上位ネットワーク６を介して、他の装置に接続されていてもよい。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３００および１または複数の表示装置４００が接続されてもよい。
【００８５】
サーバ装置３００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。情報系サービスとしては、制御対象の製造装置や設備からの情報を取得して、マクロ的またはミクロ的な分析などを行う処理が想定される。例えば、制御対象の製造装置や設備からの情報に含まれる何らかの特徴的な傾向を抽出するデータマイニングや、制御対象の設備や機械からの情報に基づく機械学習を行うための機械学習ツールなどが想定される。
【００８６】
表示装置４００は、ユーザからの操作を受けて、制御装置１００に対してユーザ操作に応じたコマンドなどを出力するとともに、制御装置１００での演算結果などをグラフィカルに表示する。
【００８７】
さらに、制御装置１００には、サポート装置２００が接続可能になっている。サポート装置２００は、制御装置１００が制御対象を制御するために必要な準備を支援する装置である。具体的には、サポート装置２００は、制御装置１００で実行されるプログラムの開発環境（プログラム作成編集ツール、パーサ、コンパイラなど）、制御装置１００および制御装置１００に接続される各種デバイスのパラメータ（コンフィギュレーション）を設定するための設定環境、生成したユーザプログラムを制御装置１００へ出力する機能、制御装置１００上で実行されるユーザプログラムなどをオンラインで修正・変更する機能、などを提供する。
【００８８】
＜Ｃ．制御装置のハードウェア構成例＞
次に、本実施の形態に係る制御装置１００のハードウェア構成例について説明する。図３は、本実施の形態に係る制御装置１００のハードウェア構成例を示すブロック図である。
【００８９】
図３を参照して、制御装置１００は、ＣＰＵユニットと称される演算処理部であり、プロセッサ１０２と、チップセット１０４と、主メモリ１０６と、ストレージ１０８と、上位ネットワークコントローラ１１０と、ＵＳＢ（Universal Serial Bus）コントローラ１１２と、メモリカードインターフェイス１１４と、内部バスコントローラ１２０と、フィールドネットワークコントローラ１３０とを含む。
【００９０】
プロセッサ１０２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１０２としては、複数のコアを有する構成を採用してもよいし、プロセッサ１０２を複数配置してもよい。すなわち、制御装置１００は、１または複数のプロセッサ１０２、および／または、１または複数のコアを有するプロセッサ１０２を有している。チップセット１０４は、プロセッサ１０２および周辺エレメントを制御することで、制御装置１００全体として

の処理を実現する。主メモリ１０６は、ＤＲＡＭ（Dynamic Random Access Memory）やＳＲＡＭ（Static Random Access Memory）などの揮発性記憶装置などで構成される。ストレージ１０８は、例えば、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。

【００９１】

プロセッサ１０２は、ストレージ１０８に格納された各種プログラムを読出して、主メモリ１０６に展開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。ストレージ１０８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３４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３０およびアプリケーションプログラム３２が格納される。

【００９２】

上位ネットワークコントローラ１１０は、上位ネットワーク６を介して、サーバ装置３００や表示装置４００（図３参照）などとの間のデータの遣り取りを制御する。ＵＳＢコントローラ１１２は、ＵＳＢ接続を介してサポート装置２００との間のデータの遣り取りを制御する。

【００９３】

メモリカードインターフェイス１１４は、メモリカード１１６が着脱可能に構成されており、メモリカード１１６に対してデータを書込み、メモリカード１１６から各種データ（ユーザプログラムやトレースデータなど）を読出すことが可能になっている。

【００９４】

内部バスコントローラ１２０は、制御装置１００に装着されるＩ／Ｏユニット１２２との間のデータの遣り取りを制御する。フィールドネットワークコントローラ１３０は、フィールドネットワーク２を介したフィールドデバイスとの間のデータの遣り取りを制御する。

【００９５】

図３には、プロセッサ１０２がプログラムを実行することで必要な機能が提供される構成例を示したが、これらの提供される機能の一部または全部を、専用のハードウェア回路（例えば、ＡＳＩＣまたはＦＰＧＡなど）を用いて実装してもよい。あるいは、制御装置１００の主要部を、汎用的なアーキテクチャに従うハードウェア（例えば、汎用パソコンをベースとした産業用パソコン）を用いて実現してもよい。この場合には、仮想化技術を用いて、用途の異なる複数のＯＳ（Operating System）を並列的に実行させるとともに、各ＯＳ上で必要なアプリケーションを実行させるようにしてもよい。

【００９６】

図２に示す制御システム１においては、制御装置１００、サポート装置２００および表示装置４００がそれぞれ別体として構成されているが、これらの機能の全部または一部を単一の装置に集約するような構成を採用してもよい。

【００９７】

＜Ｄ．制御装置の機能構成例＞

次に、本実施の形態に係る制御装置１００の機能構成例について説明する。図４は、本実施の形態に係る制御装置１００の機能構成例を示すブロック図である。

【００９８】

図４には、制御装置１００が制御アプリケーション１および制御アプリケーション２を制御する構成例を示す。制御アプリケーション１および制御アプリケーション２の各々は、典型的には、リレーやコンタクタなどのＩ／Ｏデバイスおよびサーボモータなどの各種アクチュエータを含む。制御アプリケーション１および制御アプリケーション２に加えて、他のＩ／Ｏデバイスおよび各種センサについても、フィールドネットワーク２を介して制御装置１００と接続されている。

【００９９】

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３００などから、生産の開始／終了といった指示を受ける。サーバ装置３００は、レシピ情報３８（

生産品種や生産に適したパラメータなどの情報）を制御装置１００に送信することもある。制御装置１００は、図示しない他の制御装置１００とネットワーク接続されていてもよい。

【０１００】

図４を参照して、制御装置１００は、ＰＬＣ処理エンジン１５０と、アプリケーション解析部１６０と、アプリケーション調停部１６２と、制御指令演算部１６４と、共有メモリ１７０と、共有メモリ制御部１７２と、表示機能部１７４と、フィールドネットワークインターフェイス１７６と、上位ネットワークインターフェイス１７８とを含む。

【０１０１】

ＰＬＣ処理エンジン１５０は、ユーザプログラム３０の実行および制御装置１００全体の処理を管理する。より具体的には、ＰＬＣ処理エンジン１５０は、シーケンスプログラム実行部１５２と、スケジューラ１５４と、優先度変更部１５６とを含む。

【０１０２】

シーケンスプログラム実行部１５２は、制御周期Ｔ１毎にユーザプログラム３０を実行（スキャン）して制御指令を出力する。

【０１０３】

スケジューラ１５４は、制御装置１００において実行される処理の順序を調停するものであり、具体的には、予め設定された優先度に基づいて、１または複数のタスクに対してプロセッサリソースを割当てる。

【０１０４】

優先度変更部１５６は、アプリケーション解析部１６０における処理状況を監視するとともに、アプリケーション解析部１６０における処理状況が予め定められた条件を満たすと、アプリケーション解析部１６０の処理を含むタスクに設定されている優先度を当該条件に応じて変更する。優先度変更部１５６は、アプリケーション解析部１６０による内部コマンドの生成が余裕をもって実行されるように、優先度変更部１５６に対するプロセッサリソースの割当てを調整する。

【０１０５】

優先度変更部１５６は、優先度を変更するための条件設定情報１５８を有している。条件設定情報１５８は、後述するような、条件および当該条件が満たされた場合の優先度変更処理の内容を含む。

【０１０６】

アプリケーション解析部１６０は、アプリケーションプログラム３２の少なくとも一部を解析して、内部コマンド４０を生成する。アプリケーション解析部１６０には、異なる種類のアプリケーションプログラム３２が提供されることもある。アプリケーション解析部１６０は、予め格納されたアプリケーションプログラム３２を実行するようにしてもよいし、サーバ装置３００から適宜送信されるアプリケーションプログラム３２を実行するようにしてもよい。

【０１０７】

制御指令演算部１６４は、ユーザプログラム３０に含まれるモーション命令に従って、制御指令を制御周期Ｔ１毎に演算する。また、制御指令演算部１６４は、アプリケーション解析部１６０により逐次生成される内部コマンド４０に従って、制御指令を制御周期Ｔ１毎に演算する。

【０１０８】

シーケンスプログラム実行部１５２および制御指令演算部１６４は、制御周期Ｔ１毎に処理を実行する（高優先タスク）。一方、アプリケーション解析部１６０によるアプリケーションプログラム３２に対する処理は、制御周期Ｔ１の整数倍であるアプリケーション実行周期Ｔ２（第２の周期）毎に実行される（低優先タスク）。

【０１０９】

アプリケーション調停部１６２は、アプリケーション解析部１６０において複数のアプリケーションプログラム３２が処理される場合に、その処理順序などを調停する。

【０１１０】
共有メモリ１７０は、ＰＬＣ処理エンジン１５０、アプリケーション解析部１６０、および制御指令演算部１６４の間で共有されるデータを保持する。共有されるデータは、構造体変数として格納されるようにしてもよい。例えば、アプリケーション解析部１６０により逐次生成される内部コマンド４０は、共有メモリ１７０の構造体変数に逐次書込まれる。
【０１１１】
共有メモリ制御部１７２は、共有メモリ１７０に対するデータ読書きの排他制御、および、外部からの要求に応じた共有メモリ１７０へのアクセスなどを行う。例えば、共有メモリ制御部１７２は、共有メモリ１７０上のデータをフィールドネットワークインターフェイス１７６へ与えることで、フィールドネットワーク２を介して接続される任意のフィールド機器５００へ送信される。
【０１１２】
表示機能部１７４は、共有メモリ１７０に格納されるデータおよびアプリケーション解析部１６０による処理結果などをユーザなどへ出力する。
【０１１３】
フィールドネットワークインターフェイス１７６は、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５００との間のデータの遣り取りを仲介する。
【０１１４】
上位ネットワークインターフェイス１７８は、上位ネットワーク６を介して接続されている装置との間のデータの遣り取りを仲介する。
【０１１５】
＜Ｅ．内部コマンドの一例＞
次に、制御装置１００のアプリケーション解析部１６０がアプリケーションプログラム３２を解析して生成する内部コマンド４０の一例について説明する。アプリケーションプログラム３２としては、インタプリタ方式で実行可能な任意の言語で記述されたどのようなプログラムを採用することもできるが、以下の説明においては、ＮＣプログラムまたはロボットプログラムのように、１または複数のコマンドのより予め軌跡を規定するプログラムを想定する。
【０１１６】
図５は、本実施の形態に係る制御装置１００における内部コマンド４０の生成処理を説明するための模式図である。図５（Ａ）を参照して、アプリケーション解析部１６０がアプリケーションプログラム３２を構文解析して、アプリケーションプログラム３２に含まれる各命令が解析される（（１）アプリケーションプログラム解析）。アプリケーションプログラム３２を構文解析によって、規定された軌跡が内部的に生成される（（２）軌跡生成）。アプリケーションプログラム３２は、区間毎に軌跡を規定する命令を含むことが多いので、各命令に対応する区間毎に軌跡が生成される。
【０１１７】
複数の軸からなるグループが規定される場合には、軸毎に軌跡を生成してもよいし、当該グループに属する軸の全体の挙動を規定する軌跡を生成してもよい。
【０１１８】
アプリケーション解析部１６０は、生成した軌跡を示す内部コマンド４０（典型的には、１または複数の関数）を生成する（（３）内部コマンド生成）。軌跡が区間毎に規定される場合には、各区間に対応する内部コマンド４０が生成される。
【０１１９】
以上のように、アプリケーション解析部１６０は、アプリケーションプログラム３２を構文解析して軌跡上の通過点を算出し、算出した通過点に基づいて内部コマンドを生成するようになっている。
【０１２０】
複数の区間を共通の内部コマンド４０で規定してもよいし、１つの区間をさらに分割し

て個別の内部コマンド４０をそれぞれ生成してもよい。すなわち、アプリケーションプログラム３２の命令または当該命令によって規定される軌跡の区間と、生成される内部コマンド４０の数とを一致させる必要はなく、任意に生成すればよい。また、内部コマンド４０の出力形態についても、要求される制御周期Ｔ１の時間幅などを考慮して、適宜設計されてもよい。

【０１２１】

図５（Ａ）に示すように、内部コマンド４０の一例としては、時間と指令値との関係を規定する関数であってもよい。図５（Ａ）に示す例では、内部的に生成された軌跡は、直線の組合せで規定できる。一例として、Ｘ軸に関して、直線区間毎（区間１～区間３）の軌跡を時間と速度との関係を示すＦｘ１（ｔ），Ｆｘ２（ｔ），Ｆｘ３（ｔ）を出力できる。同一のグループに属する他の軸（例えば、Ｙ軸およびＺ軸）についても、それぞれ同様に関数が出力されてもよい。

【０１２２】

図５（Ｂ）に示すように、制御指令演算部１６４が、制御周期Ｔ１毎に、生成された内部コマンド４０に従って制御指令を演算することで、制御周期Ｔ１毎に制御指令が出力される（（４）指令値演算）。すなわち、各区間に対応する関数に、各制御周期の時刻を入力することで、当該時刻における指令値を一意に決定できる。何らかのグループが設定されている場合には、当該グループに属する各軸について、指令値を同期して出力することが好ましい。

【０１２３】

上述の図５には、一例として、ＣＮＣで用いられるＧ言語により記述されたコマンドの一例を示すが、これに限らず、任意のインタプリタ方式で実行されるプログラムであれば、どのような言語を用いてもよい。また、処理対象の言語形式に応じて、生成される内部コマンド４０の形式を異ならせてもよい。

【０１２４】

＜Ｆ．タスクの実行タイミング＞

次に、本実施の形態に係る制御装置１００におけるそれぞれのタスクの実行タイミングについて説明する。図６は、本実施の形態に係る制御装置１００におけるタスクの実行タイミングの一例を示すタイムチャートである。

【０１２５】

図６を参照して、高優先タスクとして、第１タスク１０（入出力リフレッシュ処理）および第２タスク１８（シーケンス命令実行処理１２と、アプリケーションプログラムに従う制御指令の出力処理１４と、ユーザプログラムに含まれるモーション命令に従う制御指令の出力処理１６とを含む）が設定されている。低優先タスクとして、第３タスク２０（アプリケーションプログラム３２を構文解析して内部コマンド４０を逐次生成する処理）が設定されている。

【０１２６】

高優先タスクは、制御周期Ｔ１毎に実行される。アプリケーションプログラムに従う制御指令の出力処理１４においては、アプリケーション解析部１６０により生成される内部コマンド４０が、共有メモリ１７０から内部コマンド４０が読出されて（デキューされて）、当該制御周期Ｔ１における制御指令が算出される。

【０１２７】

低優先タスクは、アプリケーション実行周期Ｔ２毎に実行される。アプリケーション実行周期Ｔ２は、制御周期Ｔ１の整数倍（図６に示す例では、２倍）の単位で設定される。すなわち、アプリケーション解析部１６０は、アプリケーション実行周期Ｔ２毎に、アプリケーションプログラム３２の少なくとも一部を構文解析して内部コマンド４０を逐次生成する。生成される内部コマンド４０は、逐次、共有メモリ１７０内のバッファにキューイング（エンキュー）される。

【０１２８】

低優先タスクは、アプリケーション実行周期Ｔ２内に処理を完了できればよい。低優先

タスクは、高優先タスクが実行される期間中には、プロセッサリソースが割当てられないので、中断（サスペンド）状態で待機する。

【０１２９】

図６に示すようなそれぞれのタスクの実行タイミング、すなわちプロセッサリソースの割当ては、スケジューラ１５４によって実行される。スケジューラ１５４は、各タスクに設定されている優先度に基づいて、各タスクに対するプロセッサリソースの割当てを行う。

【０１３０】

＜Ｇ．優先度変更＞

次に、低優先タスクに設定される優先度の変更処理について説明する。上述の図１に示したように、高優先タスク以外の複数のタスクが並列的に実行されており、これらのタスクの間では、優先度に応じて、プロセッサリソースが割当てられる。

【０１３１】

このような低優先タスクは、他の低優先タスクとの間でプロセッサリソースを共有することになる。一方で、アプリケーションプログラム３２はインタプリタ方式のコマンドを含むため、構文解析に要する時間は、コマンドの種類および組合せによって変化する。そのため、要求される構文解析の処理をアプリケーション実行周期Ｔ２内に処理を完了できない場合もある。このようなアプリケーション実行周期Ｔ２内での処理を完了できない状態を、「タスク実行周期超過」とも称す。

【０１３２】

図７は、本実施の形態に係る制御装置１００における低優先タスクのタスク実行周期超過の一例を示すタイムチャートである。図７を参照して、時刻ｔ１から始まるアプリケーション実行周期Ｔ２において開始された第３タスク２０は、本来の周期の終わりである時刻ｔ３までに処理を完了することができず、次のアプリケーション実行周期Ｔ２において処理が完了している。

【０１３３】

このようなタスク実行周期超過が生じた場合には、アプリケーション実行周期Ｔ２を長くする、および／または、より高い優先度に変更する、といった対処が必要となる。

【０１３４】

本実施の形態に係る制御装置１００は、第３タスク２０（アプリケーションプログラム３２に対する構文解析処理）における処理状況を監視するとともに、当該処理状況が予め定められた条件を満たすと、第３タスク２０に設定されている優先度を当該条件に応じて変更する。この条件に応じた変更は、優先度を上げる、および、優先度を下げる、のいずれをも含み得る。

【０１３５】

典型的には、制御装置１００の優先度変更部１５６は、上述したようなタスク実行周期超過が生じた場合には、第３タスク２０に設定されている優先度を上げ、その優先度を上げた後に、予め定められた別の条件が満たされると、当該優先度を戻してもよい。

【０１３６】

優先度を変更するための「タスクの処理状況」とは、高優先タスクとして設定されている、内部コマンドを用いた制御指令の算出処理へ影響を与え得る状況を包含する概念である。「タスクの処理状況」は、アプリケーションプログラム３２に従う制御指令の制御周期Ｔ１毎の出力を保証できなくなる事態を回避するために必要な任意の情報を含み得る。

【０１３７】

制御装置１００の優先度変更部１５６は、アプリケーション解析部１６０による内部コマンドの生成処理に係る負荷を示す情報に基づいて、アプリケーション解析部１６０における処理状況を判断する。内部コマンドの生成処理に係る負荷を示す情報の一例として、以下のような情報が挙げられる。

【０１３８】

（１）低優先タスクの処理実行に要する時間

（２）タスク実行周期超過の発生有無
（３）内部コマンドを生成するための通過点のバッファ数
（４）内部コマンドのバッファ数
（５）アプリケーションプログラム内の特殊コマンド

以下、各情報に基づく優先度変更処理について説明する。

【０１３９】

（ｇ１．低優先タスクの処理実行に要する時間）

上述したタスクの処理状況として、対象となる低優先タスクの処理実行に要する時間に基づいてもよい。このように、内部コマンドの生成処理に係る負荷を示す情報は、アプリケーション解析部１６０によるアプリケーションプログラム３２の構文解析に要する時間を含んでいてもよい。

【０１４０】

低優先タスクの処理実行に要する時間としては、低優先タスクが処理開始してから処理完了までに要した時間の直前値、最大値、最小値、平均値、中間値などを用いることができる。

【０１４１】

また、対象の低優先タスクにおいてタスク実行周期超過が生じているか否か、および、タスク実行周期超過が生じた回数などを用いてもよい。

【０１４２】

図８は、本実施の形態に係る制御装置１００における処理実行に要する時間を示す変数リストの一例を示す図である。図８を参照して、制御装置１００は、システム変数として、特定のタスクの処理実行に要する情報を保持している。

【０１４３】

図８に示す例では、システム変数群６００として、タスク実行回数変数６０１（_CNC_ServiceExecCount）と、実行時間直前値変数６０２（_CNC_ServiceLastExecTime）と、実行時間最大値６０３（_CNC_ServiceMaxExecTime）と、実行時間最小値６０４（_CNC_ServiceMinExecTime）とが示されている。

【０１４４】

タスク実行回数変数６０１は、制御装置１００が起動してから（あるいは、任意のリセットタイミングから）対象タスクが実行された総回数を示す。実行時間直前値変数６０２は、対象タスクの直前の実行において処理に要した時間を示す。実行時間最大値６０３は、制御装置１００が起動してから（あるいは、任意のリセットタイミングから）のタスクの実行に要した時間の最大値を示す。実行時間最小値６０４は、制御装置１００が起動してから（あるいは、任意のリセットタイミングから）のタスクの実行に要した時間の最小値を示す。

【０１４５】

このような低優先タスクの処理実行に要する時間に基づいて、対象タスクの負荷状況などを評価し、その評価結果に基づいて、当該対象タスクの優先度を変更できる。

【０１４６】

図８に示すような処理実行に要する時間を用いる場合の条件としては、例えば、対象タスク処理実行の直前値が、予め設定されているアプリケーション実行周期Ｔ２の所定比率（例えば、９０％）まで到達した場合などを採用できる。この条件が満たされると、当該対象タスクの優先度を上げるようにしてもよい。

【０１４７】

逆に、対象タスク処理実行の直前値が、複数回に亘って、予め設定されているアプリケーション実行周期Ｔ２の所定比率（例えば、３０％）以下である場合などを採用できる。この条件が満たされると、当該対象タスクの優先度を下げるようにしてもよい。

【０１４８】

また、対象タスクの優先度をどの程度まで上げるのかについても、アプリケーション実行周期Ｔ２の長さと、対象タスク処理実行に要した時間との差分または比率に基づいて決

定してもよい。
【０１４９】
このように、低優先タスクの処理実行に要する時間を監視することで、対象タスクの負荷状況を直接的に評価できる。
【０１５０】
（ｇ２．タスク実行周期超過の発生有無）
上述したタスクの処理状況として、対象となる低優先タスクにおけるタスク実行周期超過の発生有無に基づいてもよい。すなわち、内部コマンドの生成処理に係る負荷を示す情報は、アプリケーション解析部１６０によるアプリケーションプログラム３２の構文解析に要する時間がアプリケーション実行周期Ｔ２（第２の周期）の長さを超過したか否かの情報を含んでいてもよい。
【０１５１】
再度図８を参照して、システム変数群６００は、タスク実行周期超過有無変数６０５（_CNC_ServiceExceeded）と、タスク実行周期超過回数変数６０６（_CNC_ServiceExceedCount）とがさらに示されている。タスク実行周期超過有無変数６０５は、対象タスクの直前の実行において、タスク実行周期超過が発生したか否かを示す。タスク実行周期超過回数変数６０６は、制御装置１００が起動してから（あるいは、任意のリセットタイミングから）のタスク実行周期超過が発生した総回数を示す。
【０１５２】
このようなタスク実行周期超過の発生有無に基づいて、対象タスクの負荷状況などを評価し、その評価結果に基づいて、当該対象タスクの優先度を変更できる。タスク実行周期超過の発生有無の情報を用いる場合には、対象タスクの直前の実行において、タスク実行周期超過が生じたことを条件として、当該対象タスクの優先度を上げるようにしてもよい。
【０１５３】
逆に、予め定められた実行回数に亘って、タスク実行周期超過が生じていないことを条件として、当該対象タスクの優先度を下げるようにしてもよい。
【０１５４】
このように、タスク実行周期超過の発生有無を監視することで、対象タスクの負荷状況を直接的に評価できる。
【０１５５】
（ｇ３．内部コマンドを生成するための通過点のバッファ数）
上述したタスクの処理状況として、アプリケーションプログラム３２を構文解析して算出される目標軌跡上の通過点のバッファ数に基づいてもよい。すなわち、内部コマンドの生成処理に係る負荷を示す情報は、アプリケーション解析部１６０が内部コマンドを生成するために事前に算出した通過点の数を含んでいてもよい。
【０１５６】
ここで、アプリケーションプログラム３２を構文解析して内部コマンド４０を生成する処理について説明する。図９は、本実施の形態に係る制御装置１００における内部コマンド４０の生成処理の一例を説明するための模式図である。図９（Ａ）には、時刻ｔ０から時刻ｔ１までの期間における内部コマンド４０を生成する処理手順を示し、図９（Ｂ）には、時刻ｔ１から時刻ｔ２までの期間における内部コマンド４０を生成する処理手順を示す。
【０１５７】
図９（Ａ）を参照して、アプリケーション解析部１６０は、アプリケーションプログラム３２を構文解析して目標軌跡を逐次決定する。アプリケーション解析部１６０は、逐次決定される目標軌跡に対して、予め定められた期間毎の通過点を逐次算出する。それぞれの通過点は、目標軌跡上の移動距離および目標軌跡上の移動速度に基づいて算出される。
【０１５８】
図９（Ａ）に示す例において、時刻ｔ０における初期位置である通過点Ｐ０（ｔ０）に

加えて、時刻ｔ１における通過点Ｐ１（ｔ１）、時刻ｔ２における通過点Ｐ２（ｔ２）、時刻ｔ３における通過点Ｐ３（ｔ３）が算出されたとする。

【０１５９】

アプリケーション解析部１６０は、少なくとも、通過点Ｐ０（ｔ０）および通過点Ｐ１（ｔ１）の情報に基づいて、通過点Ｐ０（ｔ０）から通過点Ｐ１（ｔ１）までの移動経路４４を算出する。アプリケーション解析部１６０は、算出した移動経路４４から内部コマンドＦｘ１（ｔ）を生成する。なお、図９（Ａ）には、内部コマンドＦｘ１（ｔ）のみを示すが、実際には、同時に制御しなければならないモータの数だけ内部コマンド４０が生成されることになる。

【０１６０】

移動経路４４の算出には、通過点Ｐ０（ｔ０）および通過点Ｐ１（ｔ１）の情報に加えて、通過点Ｐ１（ｔ１）に引き続く１または複数の通過点の情報を反映してもよい。より多くの通過点の情報を参照することで、内部コマンドの生成精度を高めることができる。

【０１６１】

図９（Ｂ）を参照して、時刻ｔ１から時刻ｔ２までの期間について、アプリケーション解析部１６０は、アプリケーションプログラム３２をさらに構文解析してさらに先の目標軌跡を決定する。アプリケーション解析部１６０は、決定されたさらに先の目標軌跡に対して、新たな通過点Ｐ４（ｔ４）を算出する。

【０１６２】

そして、アプリケーション解析部１６０は、通過点Ｐ１（ｔ１）および通過点Ｐ２（ｔ２）の情報に基づいて、通過点Ｐ１（ｔ１）から通過点Ｐ２（ｔ２）までの移動経路４４を算出する。アプリケーション解析部１６０は、算出した移動経路４４から内部コマンドＦｘ２（ｔ）を生成する。なお、図９（Ｂ）には、内部コマンドＦｘ２（ｔ）のみを示すが、実際には、同時に制御しなければならないモータの数だけ内部コマンド４０が生成されることになる。

【０１６３】

以上のような処理手順を繰返すことで、目標軌跡を実現するための内部コマンド４０が逐次生成される。

【０１６４】

図１０は、本実施の形態に係る制御装置１００における内部コマンド４０を生成するための通過点の処理例を示す模式図である。図１０を参照して、アプリケーション解析部１６０においては、通過点バッファ５０が用意されている。通過点バッファ５０には、逐次算出される通過点の情報（典型的には、座標値）がその算出順序に従ってキューイング（エンキュー）される。通過点バッファ５０からは、算出順序に従って、通過点の情報が読出されて（デキューされて）、対応する内部コマンド４０が逐次生成される。

【０１６５】

図１０に示す通過点バッファ５０に格納されている通過点の情報の数を「通過点バッファ数」とも称す。通過点バッファ数は、内部コマンド４０を逐次生成するための尤度を示すものである。すなわち、通過点バッファ数が多いほど、アプリケーション解析部１６０によるアプリケーションプログラム３２の構文解析処理において、処理時間のバラツキが発生したとしても、内部コマンド４０の生成処理が中断されることなく、継続させることができる。

【０１６６】

このようなアプリケーション解析部１６０の通過点バッファ５０に格納される通過点の数に基づいて、対象タスクの負荷状況などを評価し、その評価結果に基づいて、当該対象タスクの優先度を変更できる。

【０１６７】

図１０に示すような通過点バッファ５０に格納されている通過点の情報の数（通過点バッファ数）を用いる場合には、例えば、通過点バッファ数が、予め定められたしきい値を下回るか否かといった条件を採用できる。この条件が満たされると、当該対象タスクの優

先度を上げるようにしてもよい。
【０１６８】
逆に、通過点バッファ数が、所定期間に亘って、予め定められたしきい値を超えている状態が継続した場合には、当該対象タスクの優先度を下げるようにしてもよい。
【０１６９】
このように、アプリケーション解析部１６０の通過点バッファ５０に格納される通過点の数を監視することで、対象タスクの負荷状況を評価できる。
【０１７０】
（ｇ４．内部コマンドのバッファ数）
上述したタスクの処理状況として、アプリケーションプログラム３２から生成される内部コマンド４０のバッファ数に基づいてもよい。すなわち、内部コマンドの生成処理に係る負荷を示す情報は、アプリケーション解析部１６０により生成された内部コマンドのうち、制御指令演算部１６４により未だ処理されていない内部コマンドの数を含んでいてもよい。
【０１７１】
制御装置１００の制御指令演算部１６４は、アプリケーション解析部１６０により逐次生成される内部コマンド４０に従って、制御周期Ｔ１毎に制御指令を演算する。
【０１７２】
図１１は、本実施の形態に係る制御装置１００における内部コマンド４０から制御指令を演算する処理例を示す模式図である。図１１を参照して、共有メモリ１７０においては、アプリケーション解析部１６０により逐次生成される内部コマンドを格納するための内部コマンドバッファ６０が用意されている。内部コマンドバッファ６０には、逐次算出される内部コマンドがその生成順序に従ってキューイング（エンキュー）される。内部コマンドバッファ６０からは、生成順序に従って、内部コマンドが読出されて（デキューされて）、制御指令演算部１６４が制御指令の演算に使用する。
【０１７３】
図１１に示す内部コマンドバッファ６０に格納されている内部コマンド４０の数を「内部コマンドバッファ数」とも称す。内部コマンドバッファ数は、制御指令演算部１６４が制御指令を制御周期Ｔ１毎に演算するための尤度を示すものである。すなわち、内部コマンドバッファ数が多いほど、アプリケーション解析部１６０による内部コマンド４０の生成処理において、処理時間のバラツキが発生したとしても、制御指令の制御周期Ｔ１毎の演算処理が中断されることなく、制御指令の出力を継続させることができる。
【０１７４】
このような内部コマンドバッファ６０に格納される内部コマンド４０の数に基づいて、対象タスクの負荷状況などを評価し、その評価結果に基づいて、当該対象タスクの優先度を変更できる。
【０１７５】
図１１に示すような内部コマンドバッファ６０に格納されている内部コマンドの数（内部コマンドバッファ数）を用いる場合には、例えば、内部コマンドバッファ数が、予め定められたしきい値を下回るか否かといった条件を採用できる。この条件が満たされると、当該対象タスクの優先度を上げるようにしてもよい。
【０１７６】
逆に、内部コマンドバッファ数が、所定期間に亘って、予め定められたしきい値を超えている状態が継続した場合には、当該対象タスクの優先度を下げるようにしてもよい。
【０１７７】
このように、共有メモリ１７０の内部コマンドバッファ６０に格納される内部コマンドの数を監視することで、対象タスクの負荷状況を評価できる。
【０１７８】
（ｇ５．アプリケーションプログラム内の特殊コマンド）
上述したタスクの処理状況として、アプリケーションプログラム３２内に記述された特

殊コマンドを実行すると、その特殊コマンドに基づいて、対象タスクまたは関連するタスクの優先度を変更するようにしてもよい。すなわち、内部コマンドの生成処理に係る負荷を示す情報は、アプリケーションプログラム３２に明示的に記述される特殊コマンドを含んでいてもよい。

【０１７９】

図１２は、本実施の形態に係る制御装置１００において処理されるアプリケーションプログラムの一例を示す模式図である。図１２（Ａ）には、Ｇ言語で記述されたアプリケーションプログラム３２Ａの一例を示し、図１２（Ｂ）には、ロボット言語で記述されたアプリケーションプログラム３２Ｂの一例を示す。

【０１８０】

図１２（Ａ）を参照して、アプリケーションプログラム３２Ａは、「０５０」ブロックに特殊コマンド３２０を含む。特殊コマンド３２０は、内部コマンド４０の生成精度を２倍にするための命令である。アプリケーション解析部１６０が特殊コマンド３２０を実行すると、優先度変更部１５６に対してメッセージが通知される。優先度変更部１５６は、アプリケーション解析部１６０からのメッセージの内容に基づいて、アプリケーションプログラム３２Ａを処理するタスクについての優先度を変更する。

【０１８１】

同様に、図１２（Ｂ）を参照して、アプリケーションプログラム３２Ｂは、一連のコマンド群の途中に特殊コマンド３２２を含む。特殊コマンド３２２は、アプリケーションプログラム３２Ｂと、他のアプリケーションプログラムとを連係して動作させるための命令である。すなわち、特殊コマンド３２２が実行されることで、別のアプリケーションプログラムに対する処理が開始される。

【０１８２】

アプリケーション解析部１６０が特殊コマンド３２２を実行すると、優先度変更部１５６に対してメッセージが通知される。優先度変更部１５６は、アプリケーション解析部１６０からのメッセージの内容に基づいて、アプリケーションプログラム３２Ｂから新たに起動されるアプリケーションプログラムを処理するタスクについての優先度を変更する。あるいは、アプリケーションプログラム３２Ｂを処理するタスクについても優先度を変更してもよい。

【０１８３】

なお、図１２に示すアプリケーションプログラム３２Ａ，３２Ｂは、一例であり、どのような特殊コマンドを定義してもよい。

【０１８４】

このように、アプリケーション解析部１６０により処理されるアプリケーションプログラム内に特殊コマンドを明示的に規定することで、当該アプリケーションプログラムを処理するタスクの優先度を事前に変更することができ、対象タスクの将来的な負荷を推定した正確な優先度の調整を実現できる。

【０１８５】

（ｇ６．その他）

上述の説明においては、内部コマンドの生成処理に係る負荷を示す情報となり得る複数のファクタについて例示したが、これらのうちいずれか１つのファクタのみを用いるようにしてもよいし、任意の複数のファクタを組み合わせて用いてもよい。さらに、複数のファクタを組み合わせるにあたって、ファクタに応じた重み付けを適用してもよい。これらのファクタの利用形態としては、状況に応じた方法を適用できる。

【０１８６】

＜Ｈ．優先度変更に係る処理手順＞

次に、低優先タスクに設定される優先度の変更に係る処理手順について説明する。図１３は、本実施の形態に係る制御装置１００における優先度変更の処理手順を示すフローチャートである。図１３に示す各ステップは、典型的には、制御装置１００のプロセッサ１０２がシステムプログラム３４を実行することで実現される。

【０１８７】
図１３を参照して、ＰＬＣ処理エンジン１５０の優先度変更部１５６は、条件設定情報１５８に規定されている条件に関連する情報を収集する（ステップＳ１００）。そして、優先度変更部１５６は、収集した情報が条件設定情報１５８に規定されているいずれかの条件を満たすか否かを判断する（ステップＳ１０２）。
【０１８８】
収集した情報が条件設定情報１５８に規定されているいずれの条件も満たさなければ（ステップＳ１０２においてＮＯ）、ステップＳ１００以下の処理が繰返される。
【０１８９】
一方、収集した情報が条件設定情報１５８に規定されているいずれかの条件を満たすと（ステップＳ１０２においてＹＥＳ）、優先度変更部１５６は、当該満たされた条件に対応する内容に従って、対象のタスクについての優先度を変更する（ステップＳ１０４）。そして、ステップＳ１００以下の処理が繰返される。
【０１９０】
例えば、図１３に示すような一連の処理は、制御周期Ｔ１毎に繰返し実行されてもよい。
【０１９１】
＜Ｉ．複数のアプリケーションプログラム間の優先度調停＞
説明の便宜上、上述の説明においては、制御装置１００において、１つのアプリケーションプログラム３２が実行される場合について例示したが、複数のアプリケーションプログラム３２を並列的に実行させることもできる。このような場合には、各アプリケーションプログラムを処理するタスクについての優先度を変更するのみならず、それぞれのアプリケーションプログラムを処理するタスク間の優先度を相対的に変更（すなわち、優先度の調停）してもよい。
【０１９２】
図１４は、本実施の形態に係る制御装置１００における複数のアプリケーションプログラムを並列的に実行させる場合の実行タイミングの一例を示すタイムチャートである。図１４に示すタイムチャートにおいては、アプリケーション実行周期Ｔ２１毎に構文解析されるアプリケーションプログラム１と、アプリケーション実行周期Ｔ２２毎に構文解析されるアプリケーションプログラム２とが並列的に実行される例を示す。
【０１９３】
図１４を参照して、高優先タスクとして、第１タスク１０（入出力リフレッシュ処理）に加えて、シーケンス命令実行処理１２、ユーザプログラムに含まれるモーション命令に従う制御指令の出力処理１６、アプリケーションプログラム１に従う制御指令の出力処理１４－１、および、アプリケーションプログラム２に従う制御指令の出力処理１４－２が設定されている。
【０１９４】
アプリケーションプログラム１およびアプリケーションプログラム２の構文解析は、低有線タスクとして設定されている。
【０１９５】
なお、図１４に示す例では、高優先タスクおよび低優先タスクに対してそれぞれ独立したプロセッサリソースを割当てることができるようになっている。
【０１９６】
より具体的には、アプリケーション解析部１６０は、アプリケーション実行周期Ｔ２１（図１４に示す例では、制御周期の２倍）毎に、アプリケーションプログラム１の少なくとも一部を構文解析して内部コマンド４０を逐次生成する。生成される内部コマンド４０は、逐次、共有メモリ１７０－１内のバッファにキューイング（エンキュー）される。
【０１９７】
また、アプリケーション解析部１６０は、アプリケーション実行周期Ｔ２２（図１４に示す例では、制御周期の２倍）毎に、アプリケーションプログラム１の少なくとも一部を

構文解析して内部コマンド４０を逐次生成する。生成される内部コマンド４０は、逐次、共有メモリ１７０－２内のバッファにキューイング（エンキュー）される。

【０１９８】

なお、アプリケーションプログラム１およびアプリケーションプログラム２をそれぞれ処理する互いに独立した２つのアプリケーション解析部１６０を用意してもよい。

【０１９９】

このような構成においては、低優先タスクの間で、共通のプロセッサリソースを共有することになる。すなわち、低優先タスク用のプロセッサリソースは、低優先タスクにそれぞれ設定されている優先度の相対的な関係に基づいて配分されることになる。そのため、上述したような優先度の変更処理については、低優先タスクとして設定されているそれぞれのタスクに対して適用してもよいし、タスク間の優先度の相対的な関係を調整するように適合してもよい。

【０２００】

タスク間の優先度の相対的な関係を調整するとは、例えば、一方のタスクの優先度を上げる条件が満たされた場合には、当該タスクの優先度を上げてもよいが、他方のタスクの優先度を下げるようにしてもよい。すなわち、複数のタスクのうちいずれかのタスクを優先的に処理すべきとの条件が満たされると、当該対象のタスクの優先度を他のタスクの優先度より相対的に高くすればよく、その実現方法についてはどのようなものであってもよい。

【０２０１】

このように、低優先タスクとして複数のタスクが設定されている場合には、タスク間の優先度を調停するようにしてもよい。すなわち、アプリケーション解析部１６０が複数のアプリケーションプログラム３２のそれぞれについて構文解析を実行してそれぞれの内部コマンドを逐次生成する場合には、優先度変更部１５６は、複数のアプリケーションプログラム３２のそれぞれの構文解析に関連付けられた複数のタスクの間で、優先度を変更するようにしてもよい。このような優先度の調停は、アプリケーション調停部１６２および優先度変更部１５６（いずれも図４）が連係することで実現されてもよい。

【０２０２】

＜Ｊ．オーバーラップ動作＞

次に、本実施の形態に係る制御装置１００を用いた複数の制御アプリケーション間のオーバーラップ動作について説明する。

【０２０３】

図１５は、本実施の形態に係る制御装置１００が提供するオーバーラップ動作を説明するための模式図である。図１５には、例えば、図２に示すようなロボット５２０およびＣＮＣ工作機械５３０を含む構成への適用例を示す。具体的な動作として、ロボット５２０は、コンベア５４４により搬送されるワークＷをピックアップして、ＣＮＣ工作機械５３０の内部に配置する。ＣＮＣ工作機械５３０は、ワークＷに対して加工処理を実施する。そして、ロボット５２０は、ＣＮＣ工作機械５３０の内部からワークＷを取り出して、次工程の受入れ位置へ当該ワークＷを配置する。

【０２０４】

このような一連の動作において、生産性を向上させるためには、例えば、ＣＮＣ工作機械５３０でのワークＷに対する加工が終了する前に、ロボット５２０をワークＷの取出位置まで移動させておくことが好ましい。すなわち、ＣＮＣ工作機械５３０およびロボット５２０を、ある関係を維持した状態で、並列的に動作させる必要がある。このような複数の制御アプリケーションをある関係を維持した状態で動作させることを、以下では「オーバーラップ動作」とも称す。

【０２０５】

より具体的には、「オーバーラップ動作」は、あるアプリケーションプログラムに従う動作が開始した後に、別のアプリケーションプログラムに従う動作が後追いで開始される状態を含む。別のアプリケーションプログラムに従う動作を開始する条件（以下、「オー

バーラップ条件」とも称す。）は、動作開始からの予め定められた時間の経過、および、所定位置への到達などを含む。

【０２０６】

図１５には、ＣＮＣ工作機械５３０においてワークＷに対する加工が終了する直前の動作例を示す。ＣＮＣ工作機械５３０のワークＷに対する加工処理が時刻ｔ２に終了するとすれば、ロボット５２０は、時刻ｔ２において加工処理が終了したワークＷを取り出すことが好ましい。そのため、ロボット５２０は、時刻ｔ２において、ＣＮＣ工作機械５３０のワーク取出位置に到着できるように、時刻ｔ１から動作を開始する。そして、ロボット５２０は、時刻ｔ２においてワークＷを取り出し、時刻ｔ３から次工程への移動を開始する。一方、ＣＮＣ工作機械５３０は、ワークＷがロボット５２０により取り出されるとすぐに、新たなワークＷを受入れるための受入位置までの移動を開始する。

【０２０７】

このように、ＣＮＣ工作機械５３０およびロボット５２０を互いに連係させて、並列的に動作させることで、生産性を高めることができる。

【０２０８】

以下、このようなオーバーラップ動作を実現するための構成および処理について説明する。なお、本明細書における「オーバーラップ動作」は、複数の制御アプリケーションが任意のタイミングを基準にして同時に動作を開始する場合を含み得る。

【０２０９】

図１６は、本実施の形態に係る制御装置１００におけるオーバーラップ動作を実現するためのプログラム例を示す模式図である。図１６に示すユーザプログラム３０は、図２に示す、コンベア５４４、ＣＮＣ工作機械５３０、およびロボット５２０の制御に向けられる。

【０２１０】

より具体的には、ユーザプログラム３０は、コンベア５４４を駆動するサーボモータ５４２を制御するためのモーション命令を規定するファンクションブロック３０２と、ＣＮＣ工作機械５３０を制御するためのアプリケーションプログラム３２－１（ＮＣプログラム）の実行開始を指示するファンクションブロック３０４と、ロボット５２０を制御するためのアプリケーションプログラム３２－２（ロボットプログラム）の実行開始を指示するファンクションブロック３０６とを含む。

【０２１１】

ファンクションブロック３０２，３０４，３０６の各々は、運転開始フラグがＴＲＵＥになることで、指定された処理を実行する。

【０２１２】

アプリケーションプログラム３２－１およびアプリケーションプログラム３２－２には、オーバーラップ動作を行うための特殊コマンドが付加されている。図１６には、アプリケーションプログラム３２－１の０４９ブロック目に規定されている「Ｇ０１　Ｘ－１０　Ｙ－５０」とのコマンドに従う動作に、アプリケーションプログラム３２－２の３行目に規定されている「ＭＯＶＥ　Ｘ２００　Ｙ１００」をオーバーラップ動作させる例を示している。

【０２１３】

このようなオーバーラップ動作を実現するために、先行する動作を規定するコマンド（０４９ブロック目の「Ｇ０１　Ｘ－１０　Ｙ－５０」）の直後に、特殊コマンド３４２が規定される。特殊コマンド３４２は、「ＯｖｅｒｌａｐＭｏｖｅ」との命令に加えて、当該命令を特定するための識別情報（図１６に示す例では、引数としての「１」）が付加されている。

【０２１４】

一方、アプリケーションプログラム３２－２には、オーバーラップさせたい動作を規定するコマンド（図１６に示す例では、「ＭＯＶＥ　Ｘ２００　Ｙ１００」）の直前に、特殊コマンド３４４が規定される。特殊コマンド３４４は、「ＯｖｅｒｌａｐＳｔａｒｔ」

との命令に加えて、当該命令を特定するための識別情報（図１６に示す例では、「１」）、および、オーバーラップ条件（図１６に示す例では、「１００ｍｓ」）が付加されている。オーバーラップ条件としては、時間および位置に加えて、任意の情報を用いることができる。

【０２１５】

図１７は、時間をオーバーラップ条件とした場合のオーバーラップ動作の一例を示す模式図である。図１７を参照して、アプリケーションプログラム３２－１の０４９ブロック目に記載されたコマンドに従う動作が開始された後、オーバーラップ条件である「１００ｍｓ」の経過後に、アプリケーションプログラム３２－２の「ＭＯＶＥ　Ｘ２００　Ｙ１００」とのコマンドに従う動作が開始される。すなわち、アプリケーションプログラム３２－２に従う動作は、アプリケーションプログラム３２－１の指定された動作の開始から、オーバーラップ条件に規定された時間の経過後に開始されることとなり、両者をオーバーラップして動作させることができる。

【０２１６】

図１８は、位置をオーバーラップ条件とした場合のオーバーラップ動作の一例を示す模式図である。図１８を参照して、アプリケーションプログラム３２－１の０４９ブロック目に記載されたコマンドに従う動作が開始された後、オーバーラップ条件である所定の位置への到着（あるいは、所定の範囲内への進入）を条件として、アプリケーションプログラム３２－２の「ＭＯＶＥ　Ｘ２００　Ｙ１００」とのコマンドに従う動作が開始される。すなわち、アプリケーションプログラム３２－２に従う動作は、アプリケーションプログラム３２－１の指定された動作の開始から、オーバーラップ条件に規定された位置の条件が成立すると開始されることとなり、両者をオーバーラップして動作させることができる。

【０２１７】

図１９は、本実施の形態に係る制御装置１００においてオーバーラップ動作を実現するための機能構成を示す模式図である。図１９を参照して、制御装置１００は、２つのアプリケーション解析部１６０－１，１６０－２を有している。アプリケーション解析部１６０－１および１６０－２は、それぞれＮＣプログラムおよびロボットプログラムを処理する。

【０２１８】

アプリケーション解析部１６０－１，１６０－２がアプリケーションプログラムを構文解析して生成される内部コマンドは、共有メモリ１７０に逐次格納される。制御指令演算部１６４は、共有メモリ１７０に逐次格納されるそれぞれの内部コマンドを逐次読出して、制御指令１および制御指令２を出力する。

【０２１９】

制御指令演算部１６４は、読出した内部コマンドの処理にどの程度の時間を要するのかを判断できる。例えば、各内部コマンドは、ある時間範囲についての時間と制御指令との関係を規定する関数であり、このような関数に入力可能な時間範囲を参照することで、指定された目的位置に到達するのに要する時間を算出できる。

【０２２０】

制御指令演算部１６４は、現在実行中の内部コマンドの処理に要する時間、すなわち現在位置から指定された目的位置までの移動に要する移動時間を逐次算出することができる。制御指令演算部１６４は、逐次算出される移動時間を逐次アプリケーション解析部１６０－１，１６０－２へ出力する。

【０２２１】

アプリケーション解析部１６０－１，１６０－２は、アプリケーションプログラムに含まれるコマンドをある程度余分に構文解析（すなわち、先読み（Look Ahead））しており、オーバーラップ動作を指示するための特殊コマンドの存在を予め知ることができる。

【０２２２】

また、アプリケーション解析部１６０－１とアプリケーション解析部１６０－２との間

では、オーバーラップ動作に関する情報（典型的には、オーバーラップ条件）を互いに遣り取り可能になっている。

【０２２３】

アプリケーション解析部１６０－１，１６０－２は、オーバーラップ動作を指示するための特殊コマンドが存在する行の直前まで処理が進むと、制御指令演算部１６４からの移動時間に基づいて、オーバーラップ条件を成立させることができるか否かを判断する。すなわち、アプリケーション解析部１６０－１，１６０－２は、オーバーラップ動作をどの時点で開始しなければならないのかを知ることができる。

【０２２４】

アプリケーション解析部１６０－１，１６０－２は、指定されたオーバーラップ条件に基づいてオーバーラップ動作を開始できないと事前に判断すると、優先度変更部１５６に対して、アプリケーション解析部１６０－１，１６０－２による構文解析処理を含む低優先タスクの優先度を上げるように、優先度変更部１５６へ要求（優先度変更要求）を発する。その結果、優先度変更部１５６によるアプリケーションプログラムの構文解析に対して、より多くのプロセッサリソースが割当てられることとなり、これによって、指定されたタイミングでオーバーラップ動作を開始できる。

【０２２５】

すなわち、オーバーラップ動作の開始時刻前に、必要なアプリケーションプログラムの解析処理を完了させることができる。

【０２２６】

以上のように、アプリケーションプログラムの構文解析に係るタスクの優先度を動的に変更することで、オーバーラップ動作を確実に実行できる。

【０２２７】

一般的に、ＣＮＣ工作機械を制御するためのアプリケーションプログラムおよびロボットを制御するためのアプリケーションプログラムは、１行ずつ逐次実行して実行するインタプリタ方式の言語で記述されている。このようなインタプリタ方式のプログラムは、各コマンドによって指定される移動距離などは、コマンドの種類および引数などによって変化するため、各行の処理に要する時間はコマンド毎に異なる。

【０２２８】

さらに、ＣＮＣ工作機械およびロボットをそれぞれ専用のコントローラで制御する場合いは、各コントローラの制御周期およびコントローラ間の通信周期などの影響を受けて、緻密なオーバーラップ動作を実現することは困難であった。

【０２２９】

これに対して、本実施の形態に係る制御装置１００においては、複数のアプリケーションプログラムをそれぞれ構文解析し、制御指令を制御周期毎に出力するとともに、アプリケーション間の同期動作あるいはオーバーラップ動作させることができる。

【０２３０】

このような同期動作あるいはオーバーラップ動作によって、ＣＮＣ工作機械とロボットとが連係するような生産装置をより効率的に稼動させることができ、生産能力を高めることができる。

【０２３１】

＜Ｋ．付記＞

上述したような本実施の形態は、以下のような技術思想を含む。

［構成１］

１または複数のプロセッサを有する制御装置（１００）であって、

実行毎に全体がスキャンされる第１のプログラム（３０）と、逐次実行される第２のプログラム（３２）とを格納する記憶部（１０８）と、

第１の周期毎に前記第１のプログラムを実行して第１の制御指令を出力するプログラム実行部（１５２）と、

前記第１の周期より長い第２の周期毎に前記第２のプログラムの少なくとも一部を構文

解析して内部コマンドを逐次生成する解析部（１６０）と、
前記解析部により生成された内部コマンドに従って前記第１の周期毎に第２の制御指令を出力する指令演算部（１６４）と、
予め設定された優先度に基づいて、１または複数のタスクに対してプロセッサリソースを割当てるスケジューラ（１５４）とを備え、
前記スケジューラには、少なくとも、前記プログラム実行部および前記指令演算部による処理実行を含む第１の優先度を有する第１のタスクと、前記解析部による処理実行を含む前記第１の優先度より低い第２の優先度を有する第２のタスクと、前記第１のタスクおよび前記第２のタスクとは異なる処理内容の実行を含む第３の優先度を有する第３のタスクとが設定されており、
前記解析部における処理状況を監視するとともに、前記解析部における処理状況が予め定められた条件を満たすと、前記第２のタスクに設定されている前記第２の優先度を当該条件に応じて変更する優先度変更部（１５６）とを備える、制御装置。
［構成２］
前記優先度変更部は、前記解析部による内部コマンドの生成処理に係る負荷を示す情報に基づいて、前記解析部における処理状況を判断する、構成１に記載の制御装置。
［構成３］
前記負荷を示す情報は、前記解析部による前記第２のプログラムの構文解析に要する時間（６０２；６０３；６０４）を含む、構成２に記載の制御装置。
［構成４］
前記負荷を示す情報は、前記解析部による前記第２のプログラムの構文解析に要する時間が前記第２の周期の長さを超過したか否かの情報（６０５）を含む、構成２または３に記載の制御装置。
［構成５］
前記解析部は、前記第２のプログラムを構文解析して軌跡上の通過点を算出し、算出した通過点に基づいて内部コマンドを生成するように構成されており、
前記負荷を示す情報は、前記解析部が前記内部コマンドを生成するために事前に算出した通過点の数を含む、構成２～４のいずれか１項に記載の制御装置。
［構成６］
前記負荷を示す情報は、前記解析部により生成された内部コマンドのうち、前記指令演算部により未だ処理されていない内部コマンドの数を含む、構成２～５のいずれか１項に記載の制御装置。
［構成７］
前記負荷を示す情報は、前記第２のプログラムに記述される特殊コマンドを含む、構成２～６のいずれか１項に記載の制御装置。
［構成８］
前記優先度変更部は、前記第２のタスクに設定されている前記第２の優先度を上げた後に、予め定められた別の条件が満たされると、前記第２の優先度を元に戻す、構成１～７のいずれか１項に記載の制御装置。
［構成９］
前記解析部は、複数の前記第２のプログラムのそれぞれについて構文解析を実行してそれぞれの内部コマンドを逐次生成するように構成されており、
前記優先度変更部は、前記複数の第２のプログラムのそれぞれの構文解析に関連付けられた複数の前記第２のタスクの間で、優先度を変更する、構成１～８のいずれか１項に記載の制御装置。
【０２３２】
＜Ｌ．利点＞
本実施の形態に係る制御装置は、プロセッサリソースをスケジューリングすることで、シーケンス命令およびモーション命令を含むユーザプログラムに加えて、１または複数のアプリケーションプログラムを並列的に実行できる。このような並列実行において、制御

指令を制御周期毎に出力できるので、アプリケーションプログラムに従う制御においても、ユーザプログラムに従う制御と同じ制御精度を実現できる。

【０２３３】

本実施の形態に係る制御装置は、インタプリタ方式の言語で記述された１または複数のコマンドからなるアプリケーションプログラムを構文解析して内部コマンドを生成するとともに、生成された内部コマンドに基づいて制御指令を演算する。このようなアプリケーションプログラムの構文解析の処理および制御指令の演算処理は、互いに連係して実行する必要がある。そのため、本実施の形態に係る制御装置は、アプリケーションプログラムの構文解析に係る処理の負荷状況などに応じて、当該構文解析のタスクに設定されている優先度を動的に変化させる。これによって、各コマンドの構文解析に要する時間が異なるアプリケーションプログラムを処理する場合であっても、制御指令の制御周期毎の出力を保障できる。

【０２３４】

本実施の形態に係る制御装置は、複数のアプリケーションプログラム間で、一方のアプリケーションプログラムに従う動作が開始した後、遅れて、別のアプリケーションプログラムに従う動作が開始するといった、オーバーラップ動作を実現できる。このようなオーバーラップ動作を実現するために、制御装置は、必要な内部コマンドの生成を事前に完了できるように、必要に応じて、タスクの優先度を変更する。このようなタスクの優先度の動的な変更を活用しつつ、オーバーラップ動作を実現することで、設備をより効率的に動作させることもでき、これによって生産効率を高めることもできる。

【０２３５】

今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は、上記した説明ではなく、特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

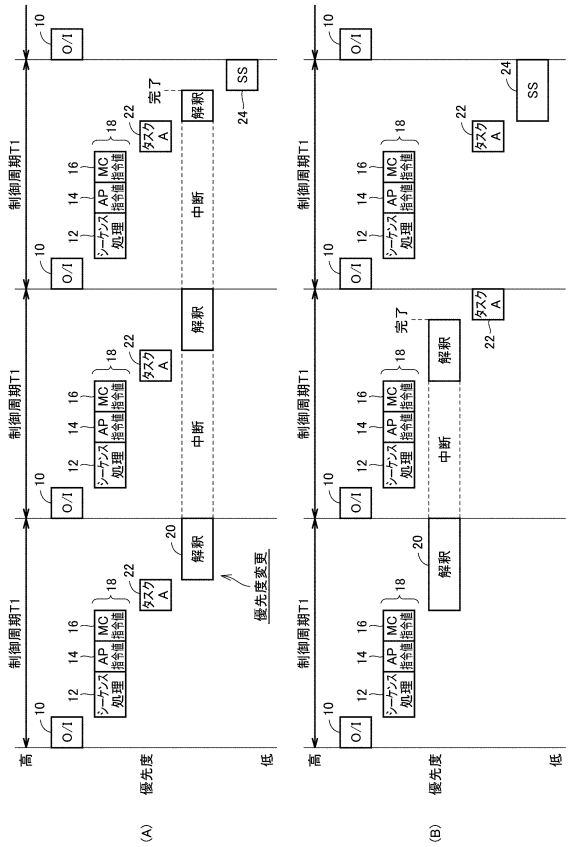
【符号の説明】

【０２３６】

１　制御システム、２　フィールドネットワーク、６　上位ネットワーク、１０　第１タスク、１２　シーケンス命令実行処理、１４，１６　出力処理、１８　第２タスク、２０　第３タスク、２２　第４タスク、２４　第５タスク、３０　ユーザプログラム、３２，３２Ａ，３２Ｂ　アプリケーションプログラム、３４　システムプログラム、３８　レシピ情報、４０　内部コマンド、４４　移動経路、５０　通過点バッファ、６０　内部コマンドバッファ、１００　制御装置、１０２　プロセッサ、１０４　チップセット、１０６　主メモリ、１０８　ストレージ、１１０　上位ネットワークコントローラ、１１２　ＵＳＢコントローラ、１１４　メモリカードインターフェイス、１１６　メモリカード、１２０　内部バスコントローラ、１２２　Ｉ／Ｏユニット、１３０　フィールドネットワークコントローラ、１５０　ＰＬＣ処理エンジン、１５２　シーケンスプログラム実行部、１５４　スケジューラ、１５６　優先度変更部、１５８　条件設定情報、１６０　アプリケーション解析部、１６２　アプリケーション調停部、１６４　制御指令演算部、１７０　共有メモリ、１７２　共有メモリ制御部、１７４　表示機能部、１７６　フィールドネットワークインターフェイス、１７８　上位ネットワークインターフェイス、２００　サポート装置、３００　サーバ装置、３０２，３０４，３０６　ファンクションブロック、３２０，３２２，３４２，３４４　特殊コマンド、４００　表示装置、５００　フィールド機器、５１０　リモートＩ／Ｏ装置、５２０　ロボット、５２２　ロボットコントローラ、５３０　工作機械、５４０　サーボドライバ、５４２　サーボモータ、５４４　コンベア、５４６　ワークテーブル、６００　システム変数群、６０１　タスク実行回数変数、６０２　実行時間直前値変数、６０３　実行時間最大値、６０４　実行時間最小値、６０５　タスク実行周期超過有無変数、６０６　タスク実行周期超過回数変数、Ｐ０，Ｐ１，Ｐ２，Ｐ３，Ｐ４　通過点、Ｔ１　制御周期、Ｔ２，Ｔ２１，Ｔ２２　アプリケーション実行周期、Ｗ　ワーク。

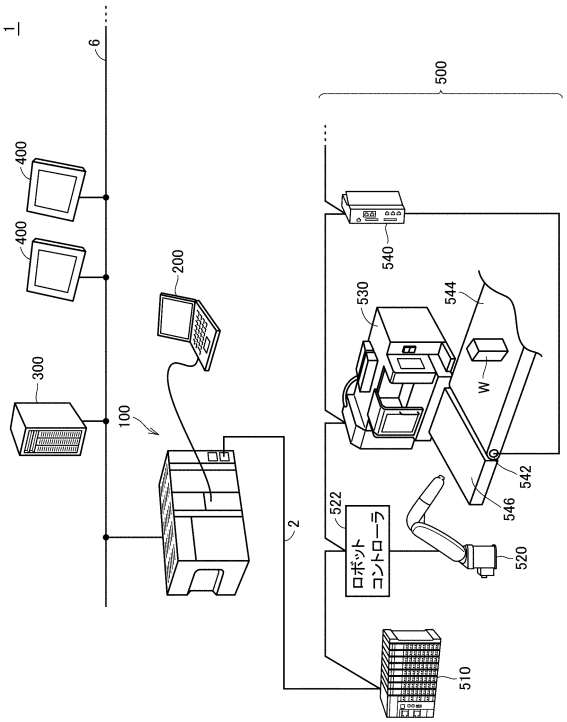
【図１】

図1



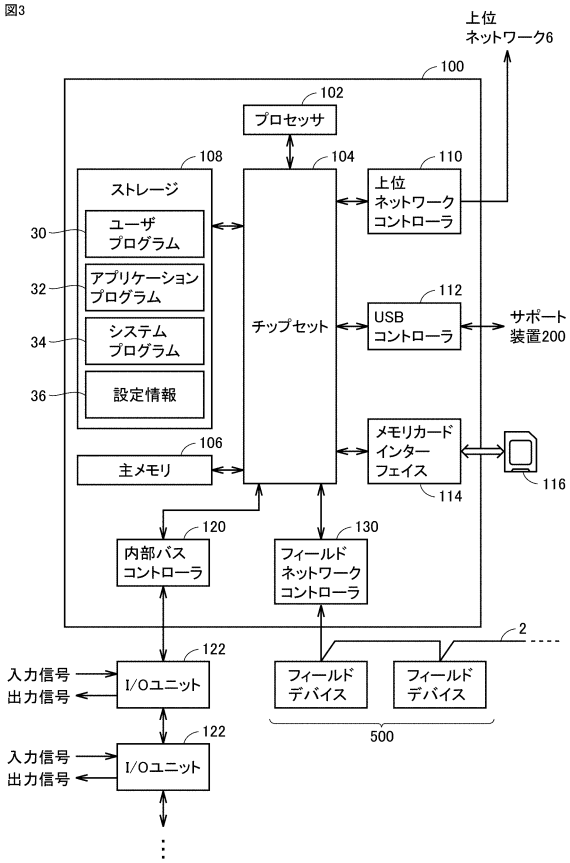
【図２】

図2



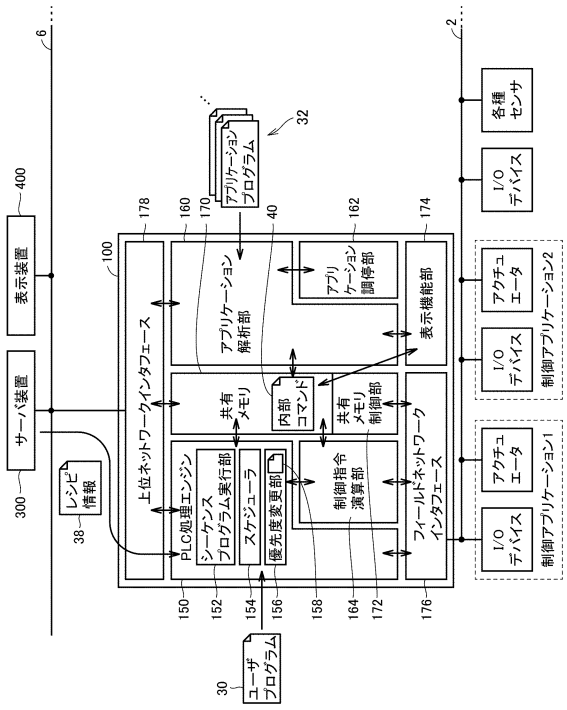
【図３】

図3



【図４】

図4



【図５】

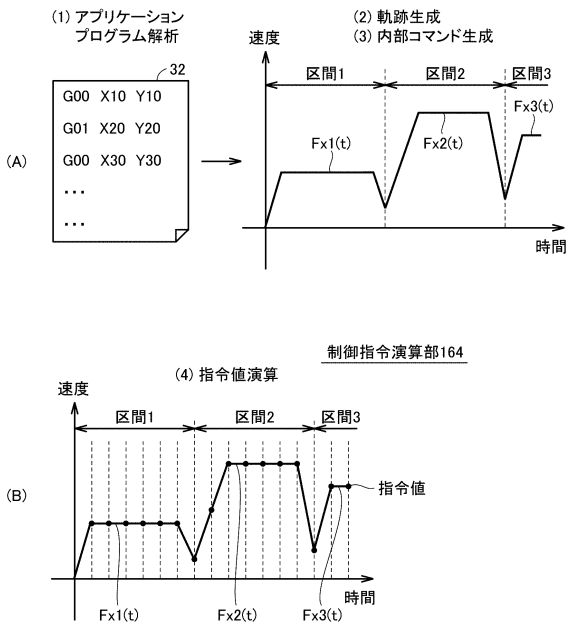


図5
(1) アプリケーション
プログラム解析
(2) 軌跡生成
(3) 内部コマンド生成
速度
32
G00 X10 Y10
G01 X20 Y20
G00 X30 Y30
(A)
区間1
区間2
区間3
Fx1(t)
Fx2(t)
Fx3(t)
時間
制御指令演算部164
(4) 指令値演算
速度
(B)
区間1
区間2
区間3
指令値
Fx1(t)
Fx2(t)
Fx3(t)
時間

【図６】

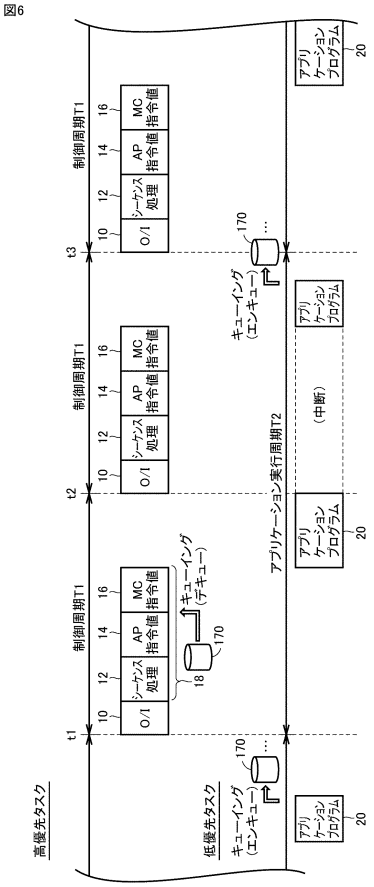


図6
制御周期T1
O/I
シーケンス処理
AP指令値
MC指令値
キューイング(デキュー)
キューイング(エンキュー)
アプリケーションプログラム
(中断)
アプリケーション実行周期T2
高優先タスク
低優先タスク
t1
t2
t3

【図７】

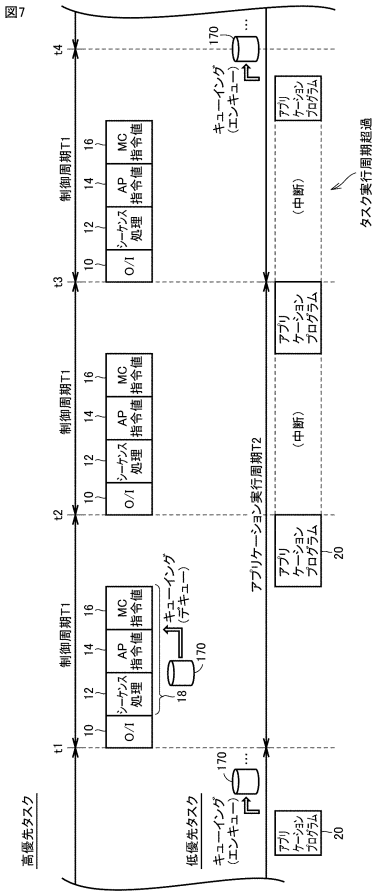


図7
制御周期T1
O/I
シーケンス処理
AP指令値
MC指令値
キューイング(デキュー)
キューイング(エンキュー)
アプリケーションプログラム
(中断)
アプリケーション実行周期T2
タスク実行周期超過
高優先タスク
低優先タスク
t1
t2
t3
t4

【図８】

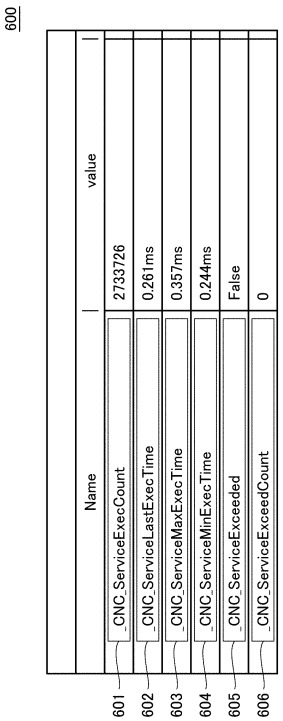
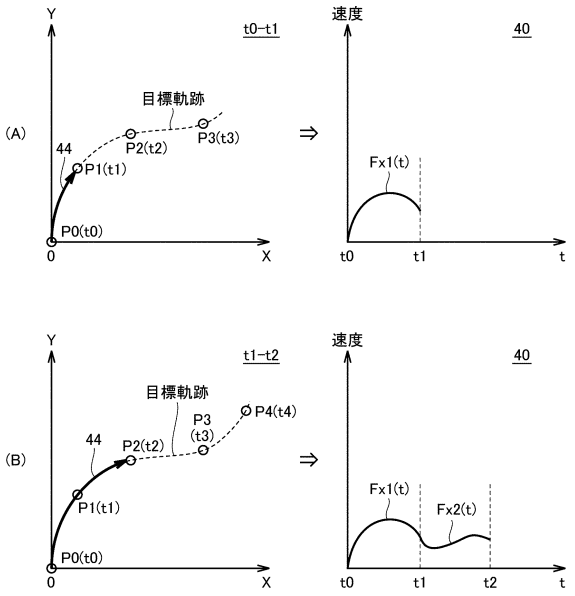


図8
600
Name
value
_CNC_ServiceExecCount
2733726
_CNC_ServiceLastExecTime
0.261ms
_CNC_ServiceMaxExecTime
0.357ms
_CNC_ServiceMinExecTime
0.244ms
_CNC_ServiceExceeded
False
_CNC_ServiceExceedCount
0
601
602
603
604
605
606

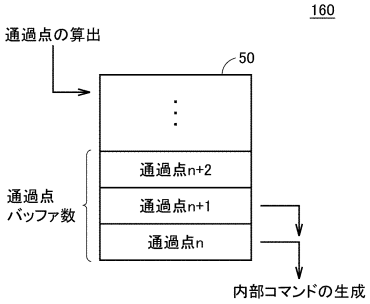
【図９】

図9



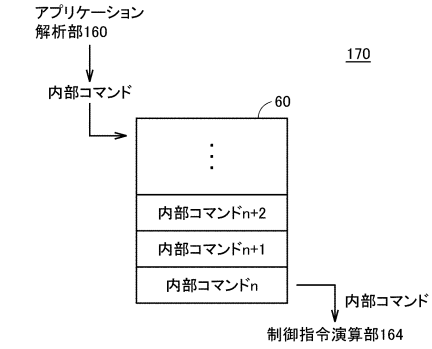
【図１０】

図10



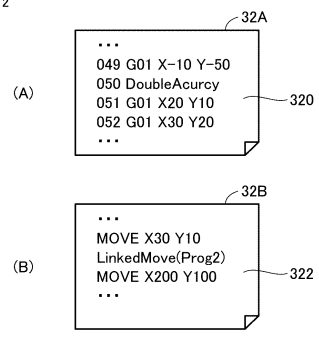
【図１１】

図11



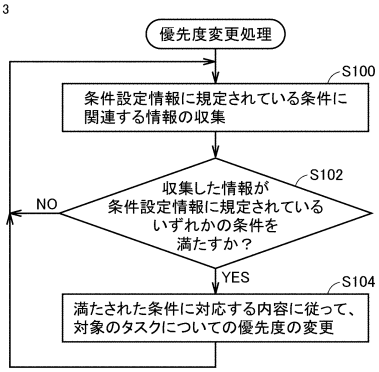
【図１２】

図12



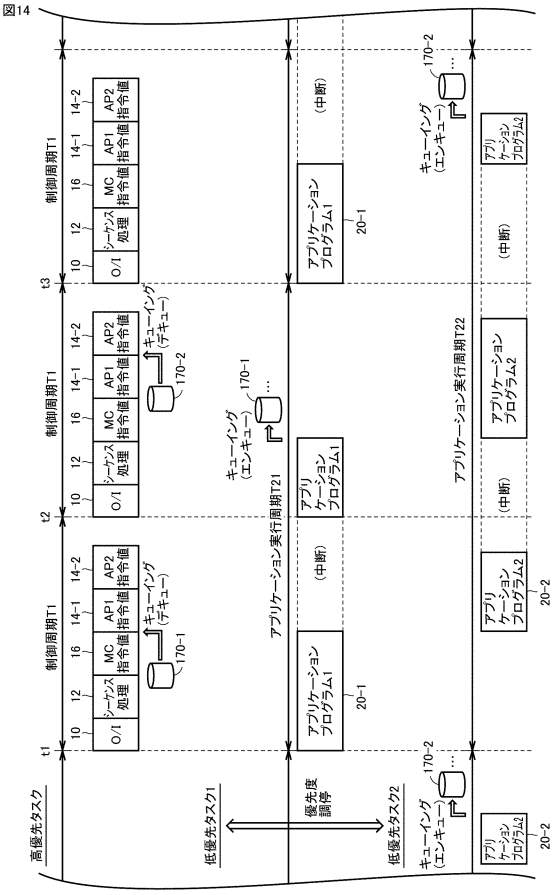
【図１３】

図13



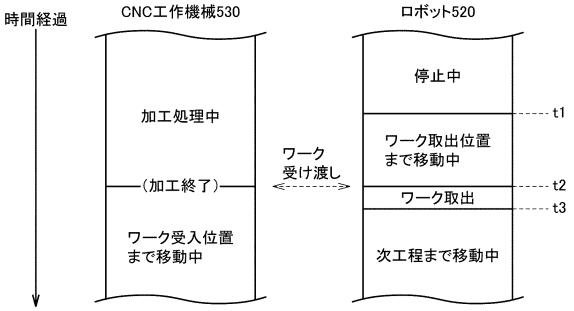
【図１４】

図14



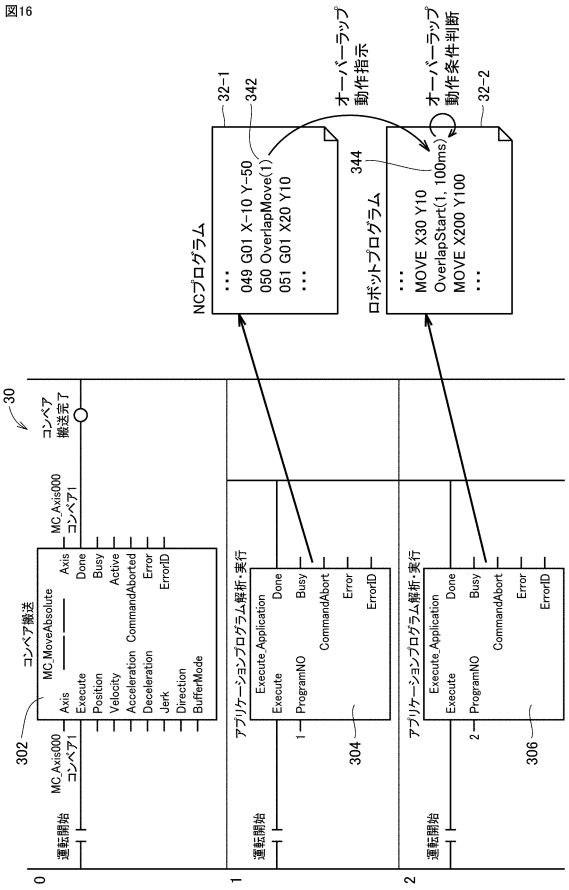
【図１５】

図15



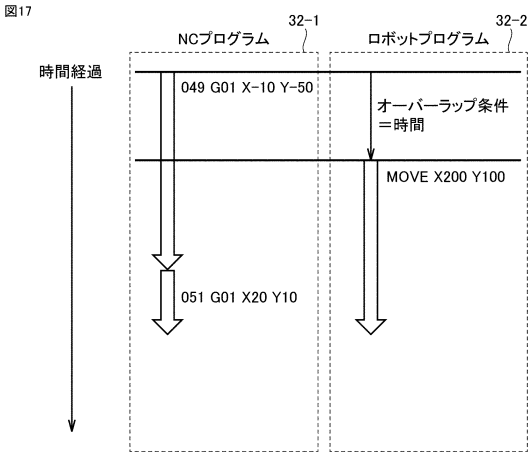
【図１６】

図16



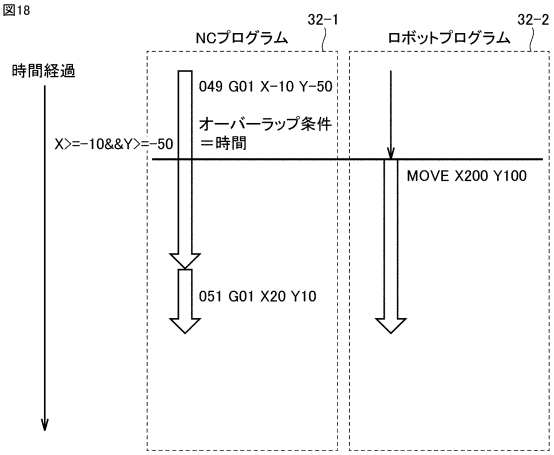
【図１７】

図17



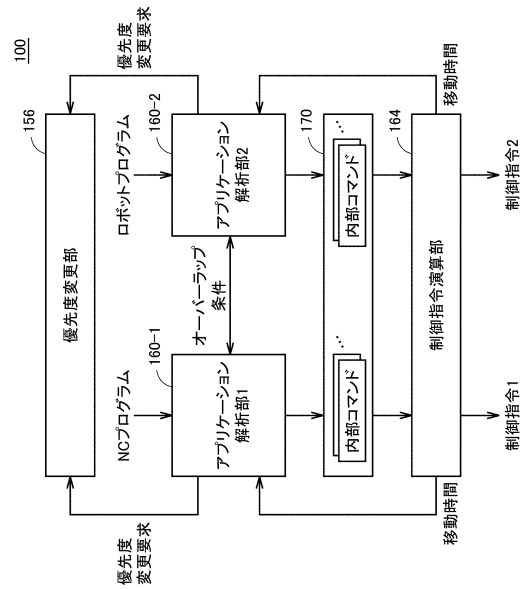
【図１８】

図18



【図１９】

図19


100
優先度変更要求
156
優先度変更部
ロボットプログラム
160-2
アプリケーション解析部2
オーバーラップ条件
NCプログラム
160-1
アプリケーション解析部1
170
内部コマンド
内部コマンド
164
制御指令演算部
移動時間
制御指令2
制御指令1
移動時間
優先度変更要求

フロントページの続き

(72)発明者　山本　英詞
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　仲野　征彦
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

審査官　漆原　孝治

(56)参考文献　特開２００１－０３４３２０（ＪＰ，Ａ）
特開２０１５－１７６１９１（ＪＰ，Ａ）
特開２００４－１５２３２２（ＪＰ，Ａ）
特開２０１６－０１２２２１（ＪＰ，Ａ）
特開２０１０－０３３１５０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４８
Ｇ０６Ｆ　　９／５０