

ロボットアーム制御機能共通インタフェース仕様書  
改版点のまとめ  
(第 1.1 版 草案)

埼玉大学工学部機械工学科 設計工学研究室

2013 年 11 月 15 日

◎改版部分を**赤色太字**にて示す

《p.1》

【改版前】

本仕様書では、6自由度あるいは7自由度を有するマニピュレータ及びその先端にエンドエフェクタとして1軸グリッパを取り付けたロボットアームを制御するための共通インタフェース(ACT インタフェース)を規定している。

【改版後】

本仕様書では、**1～3自由度の直交座標型、4自由度の水平多関節型、5～7自由度の垂直多関節型の**マニピュレータ及びその先端にエンドエフェクタとして1軸グリッパを取り付けたロボットアームを制御するための共通インタフェース(ACT インタフェース)を規定している。

《pp.12-13》

- ・直交座標の直線補間（絶対指令）

【改版前】

moveLinearCartesianAbs		RETURN_ID	ロボット座標系の絶対値で指定された目標位置に対し、直交空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	絶対目標位置・姿勢 [単位：mm、degree]
<p>「直交空間における直線補間」とは、直交空間中の各方向の並進および回転動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>6軸のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

【改版後】

moveLinearCartesianAbs		RETURN_ID	ロボット座標系の絶対値で指定された目標位置に対し、直交空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	絶対目標位置・姿勢 [単位：mm、degree]
<p>「直交空間における直線補間」とは、直交空間中の各方向の並進および回転動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p><b>1～3軸の直交座標型マニピュレータの場合は、戻り値 RETURN_ID のメンバ変数 id には NG(-1)が格納され、オペレーションは拒否される。</b></p> <p><b>4軸の水平多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における Z 軸以外の軸回転による目標姿勢は無視される。</b></p> <p><b>5軸の垂直多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における X 軸の回転による目標姿勢は無視される。</b></p> <p>また、<b>4～6軸</b>のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

・ 直交座標の直線補間 (相対指令)

【改版前】

moveLinearCartesianRel		RETURN_ID	ロボット座標系の相対値で指定された目標位置に対し、直交空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	相対目標位置・姿勢 [単位 : mm、degree]
<p>「直交空間における直線補間」とは、直交空間中の各方向の並進および回転動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>6軸のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

【改版後】

moveLinearCartesianRel		RETURN_ID	ロボット座標系の相対値で指定された目標位置に対し、直交空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	相対目標位置・姿勢 [単位 : mm、degree]
<p>「直交空間における直線補間」とは、直交空間中の各方向の並進および回転動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p><b>1～3軸の直交座標型マニピュレータの場合は、戻り値 RETURN_ID のメンバ変数 id には NG(-1)が格納され、オペレーションは拒否される。</b></p> <p><b>4軸の水平多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における Z 軸以外の軸回転による目標姿勢は無視される。</b></p> <p><b>5軸の垂直多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における X 軸の回転による目標姿勢は無視される。</b></p> <p>また、<b>4～6軸</b>のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

・ 関節座標の直線補間 (直交・絶対指令)

【改版前】

movePTPCartesianAbs		RETURN_ID	ロボット座標系の絶対値で指定された目標位置に対し、関節空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	絶対目標位置・姿勢 [単位 : mm、degree]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>6軸のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

【改版後】

movePTPCartesianAbs		RETURN_ID	ロボット座標系の絶対値で指定された目標位置に対し、関節空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	絶対目標位置・姿勢 [単位 : mm、degree]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p><b>1～3軸の直交座標型マニピュレータの場合は、戻り値 RETURN_ID のメンバ変数 id には NG(-1)が格納され、オペレーションは拒否される。</b></p> <p><b>4軸の水平多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における Z 軸以外の軸回転による目標姿勢は無視される。</b></p> <p><b>5軸の垂直多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における X 軸の回転による目標姿勢は無視される。</b></p> <p>また、<b>4～6軸</b>のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

・ 関節座標の直線補間 (直交・相対指令)

【改版前】

movePTPCartesianRel		RETURN_ID	ロボット座標系の相対値で指定された目標位置に対し、関節空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	相対目標位置・姿勢 [単位 : mm、degree]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>6軸のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

【改版後】

movePTPCartesianRel		RETURN_ID	ロボット座標系の相対値で指定された目標位置に対し、関節空間における直線補間で動作する。
in	carPoint	CarPosWithElbow	相対目標位置・姿勢 [単位 : mm、degree]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p><b>1～3軸の直交座標型マニピュレータの場合は、戻り値 RETURN_ID のメンバ変数 id には NG(-1)が格納され、オペレーションは拒否される。</b></p> <p><b>4軸の水平多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における Z 軸以外の軸回転による目標姿勢は無視される。</b></p> <p><b>5軸の垂直多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における X 軸の回転による目標姿勢は無視される。</b></p> <p>また、<b>4～6軸</b>のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

・ 関節座標の直線補間 (関節・絶対指令)

【改版前】

movePTPJntAbs		RETURN_ID	絶対関節座標系で指定された目標位置に対し、関節空間における直線補間で動作する。
in	jointPoint	JointPos	絶対目標位置 [単位 degree]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>引数 jointPoint 配列の値の順番は、J1、J2、J3、・・・とする。</p>			

【改版後】

movePTPJntAbs		RETURN_ID	絶対関節座標系で指定された目標位置に対し、関節空間における直線補間で動作する。
in	jointPoint	JointPos	絶対目標位置 [単位 degree または mm]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>引数 jointPoint 配列の値の順番は、J1、J2、J3、・・・とする。</p>			

・ 関節座標の直線補間 (関節・相対指令)

【改版前】

movePTPJntRel		RETURN_ID	相対関節座標系で指定された目標位置に対し、関節空間における直線補間で動作する。
in	jointPoint	JointPos	相対目標位置 [単位 degree]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>引数 jointPoint 配列の値の順番は、J1、J2、J3、・・・とする。</p>			

【改版後】

movePTPJntRel		RETURN_ID	相対関節座標系で指定された目標位置に対し、関節空間における直線補間で動作する。
in	jointPoint	JointPos	相対目標位置 [単位 degree または mm]
<p>「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。</p> <p>引数 jointPoint 配列の値の順番は、J1、J2、J3、・・・とする。</p>			

《p.15》

・ 直交座標の円弧補間 (絶対指令)

【改版前】

なし

【改版後】

<b>moveCircularCartesianAbs</b>		<b>RETURN_ID</b>	ロボット座標系の絶対値で指定された中継位置・目標位置に対し、直交空間における円弧補間で動作する。
<b>in</b>	<b>carPointR</b>	<b>CarPosWithElbow</b>	絶対中継位置・姿勢 [単位: mm、degree]
<b>in</b>	<b>carPointT</b>	<b>CarPosWithElbow</b>	絶対目標位置・姿勢 [単位: mm、degree]
<p>「直交空間における円弧補間」とは、直交空間中の現在位置と 2 円周点を通る円弧となるように軌跡生成する動作のことである。</p> <p>1 軸の直交座標型マニピュレータの場合は、戻り値 RETURN_ID のメンバ変数 id には NG(-1) が格納され、オペレーションは拒否される。</p> <p>2 軸の直交座標型マニピュレータの場合は、中継位置・目標位置をアーム座標系の絶対値で指定する。このとき、J1 軸を X 軸、J2 軸を Y 軸とみなし、引数 carPoint のメンバ変数 HgMatrix における Z 軸の値、及び全ての軸回転による目標姿勢は無視される。</p> <p>3 軸の直交座標型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における全ての軸の回転による目標姿勢は無視される。</p> <p>4 軸の水平多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における Z 軸以外の軸回転による目標姿勢は無視される。</p> <p>5 軸の垂直多関節型マニピュレータの場合は、引数 carPoint のメンバ変数 HgMatrix における X 軸の回転による目標姿勢は無視される。</p> <p>また、2～6 軸のマニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。</p>			

- ・ 直交座標の円弧補間 (相対指令)

【改版前】

なし

【改版後】

<b>moveCircularCartesianRel</b>		<b>RETURN_ID</b>		ロボット座標系の相対値で指定された中継位置・目標位置に対し、直交空間における円弧補間で動作する。
<b>in</b>	<b>carPointR</b>	<b>CarPosWithElbow</b>	相対中継位置・姿勢 [単位: mm、degree]	
<b>in</b>	<b>carPointT</b>	<b>CarPosWithElbow</b>	相対目標位置・姿勢 [単位: mm、degree]	
<p>「直交空間における円弧補間」とは、直交空間中の現在位置と 2 円周点を通る円弧となるように軌跡生成する動作のことである。</p> <p>1 軸の直交座標型マニピュレータの場合は、戻り値 <b>RETURN_ID</b> のメンバ変数 <b>id</b> には NG(-1) が格納され、オペレーションは拒否される。</p> <p>2 軸の直交座標型マニピュレータの場合は、中継位置・目標位置をアーム座標系の相対値で指定する。このとき、J1 軸を X 軸、J2 軸を Y 軸とみなし、引数 <b>carPoint</b> のメンバ変数 <b>HgMatrix</b> における Z 軸の値、及び全ての軸回転による目標姿勢は無視される。</p> <p>3 軸の直交座標型マニピュレータの場合は、引数 <b>carPoint</b> のメンバ変数 <b>HgMatrix</b> における全ての軸の回転による目標姿勢は無視される。</p> <p>4 軸の水平多関節型マニピュレータの場合は、引数 <b>carPoint</b> のメンバ変数 <b>HgMatrix</b> における Z 軸以外の軸回転による目標姿勢は無視される。</p> <p>5 軸の垂直多関節型マニピュレータの場合は、引数 <b>carPoint</b> のメンバ変数 <b>HgMatrix</b> における X 軸の回転による目標姿勢は無視される。</p> <p>また、2～6 軸のマニピュレータの場合は、引数 <b>carPoint</b> のメンバ変数 <b>elbow</b> は無視される。</p>				

- ・原点復帰時の位置を設定

【改版前】

なし

【改版後】

<b>setHome</b>		<b>RETURN_ID</b>	原点復帰時の位置を関節座標系の絶対値で設定する。
<b>in</b>	<b>jointPoint</b>	<b>JointPos</b>	絶対位置 [単位 : degree or mm]
本オペレーションを 1 回も実行していない場合の値は、実装依存とする。 引数 <b>jointPoint</b> 配列の値の順番は、J1、J2、J3、・・・とする。			

- ・原点復帰時の位置を設定

【改版前】

なし

【改版後】

<b>getHome</b>		<b>RETURN_ID</b>	関節座標系の絶対値で定義された原点復帰位置を取得する。
<b>out</b>	<b>jointPoint</b>	<b>JointPos</b>	絶対位置 [単位 : degree or mm]
<b>setHome</b> オペレーションで設定した値を取得する。			

- ・原点復帰動作

【改版前】

なし

【改版後】

<b>goHome</b>		<b>RETURN_ID</b>	関節座標系の絶対値で指定された原点復帰位置に対し、関節空間における直線補間で動作する。
「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。			

《p.21》

・ ManipulatorCommonInterface\_MiddleLevel.idl

【改版前】

```
#ifndef MANIPULATORCOMMONINTERFACE_MIDDLE_IDL
#define MANIPULATORCOMMONINTERFACE_MIDDLE_IDL
    中略
interface ManipulatorCommonInterface_Middle{
    RETURN_ID closeGripper();
    中略
    RETURN_ID setSpeedJoint(in ULONG spdRatio);
};
#endif // MANIPULATORCOMMONINTERFACE_MIDDLE_IDL
```

【改版後】

```
#ifndef MANIPULATORCOMMONINTERFACE_MIDDLE_IDL
#define MANIPULATORCOMMONINTERFACE_MIDDLE_IDL
    中略
interface ManipulatorCommonInterface_Middle{
    RETURN_ID closeGripper();
    中略
    RETURN_ID setSpeedJoint(in ULONG spdRatio);
    RETURN_ID moveCircularCartesianAbs(in CarPosWithElbow carPointR,
                                     in CarPosWithElbow carPointT);
    RETURN_ID moveCircularCartesianRel(in CarPosWithElbow carPointR,
                                       in CarPosWithElbow carPointT);
    RETURN_ID setHome(in JointPos jointPoint);
    RETURN_ID getHome(out JointPos jointPoint);
    RETURN_ID goHome();
};
#endif // MANIPULATORCOMMONINTERFACE_MIDDLE_IDL
```