Doc.No : 616f001
Version : E
Date : 24 Nov 2008

Algorithm defining hexapod parameters for rotating M2 around an arbitrary M2 on-axis point and performing a true relative roto-translation movement

Armando Riccardi
INAF – Osservatorio Astrofisico di Arcetri

INAF – Osservatorio Astrofisico di Arcetri
Largo E. Fermi, 5 - 50125 Firenze - ITALY
http://www.arcetri.astro.it/adopt/

ABSTRACT

The compensation of collimation coma error without affecting pointing error can be performed rotating M2 around its center of curvature. Similarly compensation pointing error without affecting collimation coma error can be performed rotating M2 around its focal point. Hexapod commands provided by the supplier allow to perform a rotation around the M2 vertex followed by a translation of the vertex itself. The present document reports the algorithms to compute the parameters of the existing hexapod command in order to perform the required rotations around an arbitrary point on the current direction of the M2 axis.

Modification Record

Version	Date	Author	Section/Paragraph affected	Reason/Remarks
A	18 Nov 2006	Armando Riccardi		First release of the document
B	03 Oct 2007	Armando Riccardi	Appendix 3	Added Appendix 3
C	10 Jan 2008	Armando Riccardi	Sec 3, Appendix III, Appendix IV	Sec. 3:The true relative positioning is now explicitly described in details Appendix III: Added formulas in C format for the true relative movement Some minor reformatting
D	12 Feb 2008	Armando Riccardi	Appendix III, Appendix IV	Added the missing formula for R1_21 term. Removed unused formula for R1_22 term.
E	24 Nov 2008	Armando Riccardi	Appendix IV	Fixed parenthesis in R1_21 equation

Abbreviations, acronyms and symbols

Symbol	Description
LBT	Large Binocular Telesecope
MRS	Main Reference System
RoC	Radius of Curvature
RRP	Relative Reference Position
SCS	Shell Coordinate System
TCS	Telescope Control System

Contents

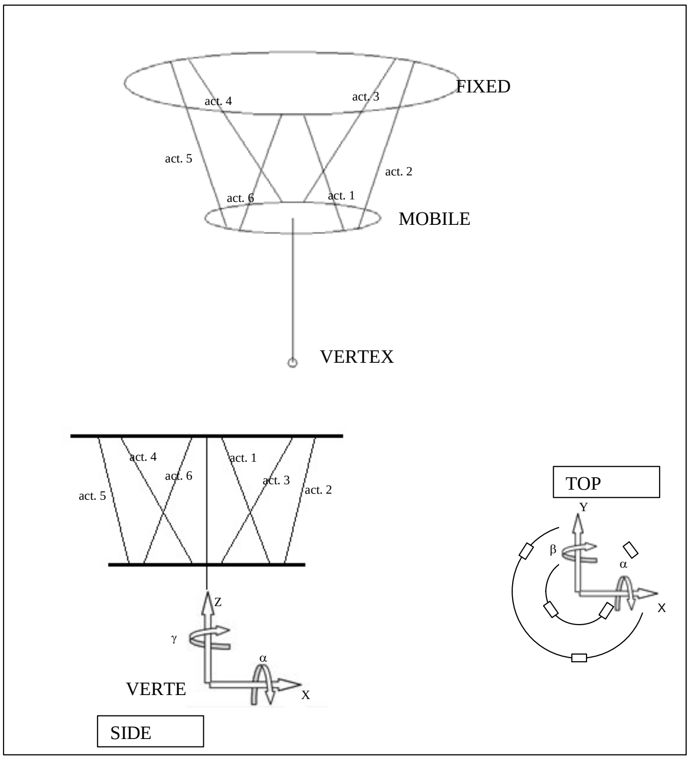

Fig. 1 Definition of Main Reference System (MRS). In the picture (from Ref. [1]) the mobile flange is in its zero position giving the M2 vertex be coincident with MRS origin O. The MRS does not move when the mobile flange moves.

1 Introduction

The compensation of collimation coma error without affecting pointing error can be performed rotating M2 around its center of curvature. Similarly compensation pointing error without affecting collimation coma error can be performed rotating M2 around its focal point. Those kinds of rotations are not implemented by the hexapod supplier in the internal kinematics of the hexapod control electronics that allows just explicit rotations around the M2 shell vertex. TCS has to compute a sequence of standard commands to guide hexapod over the required trajectory.

2 The hexapod standard internal commands

The hexapod internal kinematics integrates two kinds of moving command (see [1]):

- SET_POS_ABS: rotates the M2 shell and displaces the shell vertex with respect to the Main Reference System (MRS) that is anchored to the fixed flange.
- SET_POS_REL: rotates the M2 shell and displaces the shell vertex as incremental commands with respect to the offset commands stated by SET_REL_REF command. We will (used for setting differential commands with respect to an offset), Relative Reference System hereafter.

The O*xyz* MRS is defined in Fig. 1 (from Ref. [1]). In both cases the target position is defined by 6 parameters: α, β, γ, x_V, y_V and z_V for the SET_POS_ABS command and $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$, Δx_V, Δy_V and Δz_V for the SET_POS_REL.

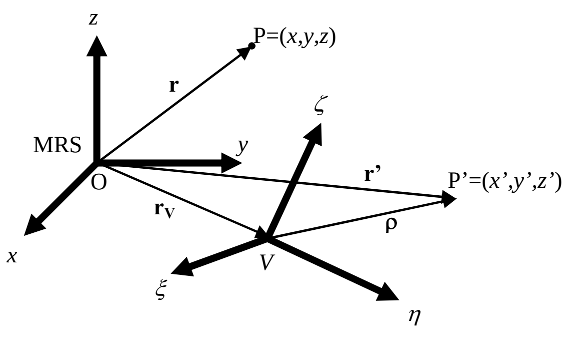

Fig. 2 Geometry of MRS O*xyz* and the reference system V$\xi\eta\zeta$ fixed with the M2 shell (V is the shell vertex).

2.1 Internal implementation of SET_POS_ABS command

For the SET_POS_ABS command the target position is computed starting from the zero position (no rotation and shell vertex V placed in the main system origin O) and applying the following sequence of transformations:

1. A rotation around the MRS x axis of an angle α;
2. A rotation around the MRS y axis of an angle β;
3. A rotation around the MRS z axis of an angle γ;
4. A rigid displacement of the M2 vertex from O to $(x,y,z)=(x_V,y_V,z_V)$. Note that the displacement of O to V is defined in terms of V coordinates in the MRS.

The above steps can be written in formulas as follow. Let's define three coordinate vectors $\mathbf{r}_V$, $\mathbf{r}$ and $\mathbf{r}'$ having components in the MRS as follow

$$\mathbf{r}_V = \begin{pmatrix} x_V \\ y_V \\ z_V \end{pmatrix}, \ \mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \text{ and } \mathbf{r}' = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}. \tag{2}$$

$\mathbf{r}_V$ is the required displacement vector of the M2 vertex V, $\mathbf{r}$ is the coordinate vector of a generic point P and $\mathbf{r}'$ is the MRS coordinate vector of the roto-translated point P' originating from the roto-translation of the point P (see Fig. 2). The relationship between the $\mathbf{r}$ and $\mathbf{r}'$ coordinates in the MRS is given by

$$\mathbf{r}' = \mathbf{R}(\alpha,\beta,\gamma)\mathbf{r} + \mathbf{r}_V, \tag{3}$$

where

$$\mathbf{R}(\alpha,\beta,\gamma) = \mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_x(\alpha) \tag{4}$$

and

$$\mathbf{R}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}, \ \mathbf{R}_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}, \ \mathbf{R}_z(\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5}$$

giving

(6) $$\mathbf{R}(\alpha,\beta,\gamma)=\begin{pmatrix}\cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma-\cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma+\sin\alpha\sin\gamma\\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma+\cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma-\sin\alpha\cos\gamma\\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta\end{pmatrix}.$$

In our case $|\alpha|$,$|\beta|$ and $|\gamma|$ are always less then 90deg (less then or equal to 0.25deg in particular). This condition allows to write the following relationships between the matrix elements R_{ij} of **R** and the rotation angles α, β and γ without any ambiguity introduced by the trigonometric inverse functions (branch selection)

(7) $$\alpha=\arctan(R_{32}/R_{33})$$

(8) $$\beta=-\arcsin(R_{31})$$

(9) $$\gamma=\arctan(R_{22}/R_{11})$$

2.2 Internal implementation of SET_POS_REL command

For the SET_POS_REL the target position is computed as an offset of the hexapod positioning parameters $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$, Δx_V, Δy_V and Δz_V of the global position parameters stored by the SET_REL_REF command.

When the SET_REL_REF command is issued, the set of parameters identifying the current position in the MRS is stored in the parameter set α_0, β_0, γ_0, x_{V0}, y_{V0} and z_{V0}. The Relative Reference Position (RRP) in MRS is given by

(10) $$\mathbf{r}_0' = \mathbf{R}(\alpha_0,\beta_0,\gamma_0)\mathbf{r}+\mathbf{r}_{V0},$$

where

(11) $$\mathbf{r}_{V0}=\begin{pmatrix}x_{V0}\\ y_{V0}\\ z_{V0}\end{pmatrix}.$$

The target position of the SET_POS_REL command issued with parameters $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$, Δx_V, Δy_V and Δz_V move the shell in the following final position in the MRS

(12) $$\mathbf{r}' = \mathbf{R}(\alpha_0+\Delta\alpha,\beta_0+\Delta\beta,\gamma_0+\Delta\gamma)\mathbf{r}+(\mathbf{r}_{V0}+\Delta\mathbf{r}_V),$$

where

(13) $$\Delta\mathbf{r}_V=\begin{pmatrix}\Delta x_V\\ \Delta y_V\\ \Delta z_V\end{pmatrix}.$$

3 How to implement a true relative positioning

The user has to be warned that the relative movement implemented by the hexapod supplier is not a proper roto-translation with respect to a Relative Reference System (RRS) axes stated by the RRP. The setting of a real RRS would give a final target position in the MRS given by

(14) $$\mathbf{r}' = \mathbf{R}(\Delta\alpha_0,\Delta\beta_0,\Delta\gamma_0)\mathbf{r}_0'+\Delta\mathbf{r}_{V0},$$

that, from Eq. (10), gives

(15) $$\mathbf{r}' = \mathbf{R}(\Delta\alpha_0,\Delta\beta_0,\Delta\gamma_0)\mathbf{R}(\alpha_0,\beta_0,\gamma_0)\mathbf{r}+[\mathbf{R}(\Delta\alpha_0,\Delta\beta_0,\Delta\gamma_0)\mathbf{r}_{V0}+\Delta\mathbf{r}_{V0}],$$

that differs from Eq. (12). See Appendix 1 for a numerical evaluation of the error that is obtained when the user would use the internal SET_POS_REL command to perform a true relative positioning movement.

Let us define

(16) $$\mathbf{R}' = \mathbf{R}(\alpha',\beta',\gamma') = \mathbf{R}(\Delta\alpha_0,\Delta\beta_0,\Delta\gamma_0)\mathbf{R}(\alpha_0,\beta_0,\gamma_0),$$

(17) $$\mathbf{r}_V' = \mathbf{R}(\Delta\alpha_0,\Delta\beta_0,\Delta\gamma_0)\mathbf{r}_{V0}+\Delta\mathbf{r}_{V0}.$$

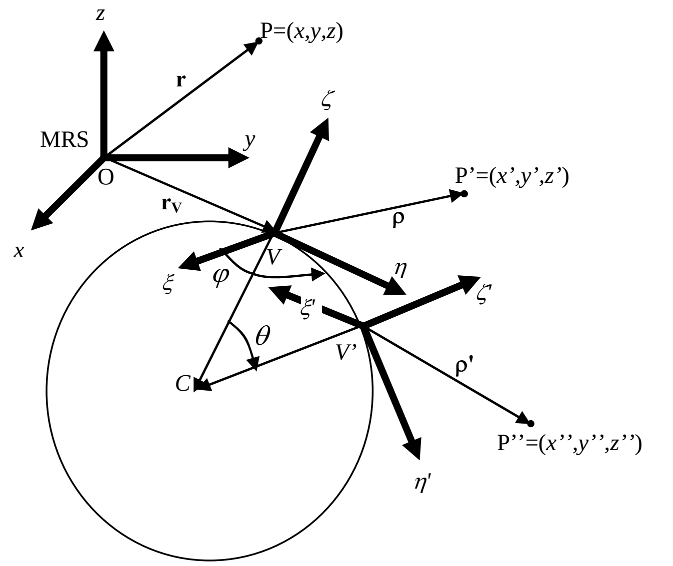

Fig. 3 Geometry of spherical rotation of the shell.

Using the above definitions, Eq. (15) can be re-written in the standard form of a MRS roto-translation (to be compared with Eq. (3))

(18) $\mathbf{r}' = \mathbf{R}'\mathbf{r} + \mathbf{r}_V{}'$.

Let's define the SET_POS_ABS parameters α', β', γ', x_V', y_V' and z_V' identifying the target absolute position. Using Eqs. (7),(8) and (9), the target position angles are given by

(19) $\alpha' = \arctan(R'_{32} / R'_{33})$

(20) $\beta' = -\arcsin(R'_{31})$

(21) $\gamma' = \arctan(R'_{21} / R'_{11})$

where R_{ij}' are the elements of the **R'** matrix defined in Eq. (16). The x_V', y_V' and z_V' parameters are given by the components of the $\mathbf{r}_V'$ vector defined in Eq. (17).

Appendix 3 reports the relevant formulas for the true-relative positioning written in C code.

4 How to implement the spherical positioning

In the spherical positioning the shell of M2 has to rotate around a point that is placed on the actual optical axis of the shell itself as shown in Fig. 3. The RoC q is positive when the rotation point is toward M1. The direction of rotation is identified by two angles φ and θ as shown in Fig. 3. The angle φ defines the direction from the ξ axis of the movement. The angle θ defines the amount of rotation from the optical axis.

Let's define the SET_POS_ABS parameters α', β', γ', x_V', y_V' and z_V' identifying the target position and the parameters α, β, γ, x_V, y_V and z_V identifying the current absolute position. The target position angles are given by

(22) $\alpha' = \arctan(R'_{32} / R'_{33})$

(23) $\beta' = -\arcsin(R'_{31})$

(24) $\gamma' = \arctan(R'_{21} / R'_{11})$

where R'_{ij} are the components of the rotation matrix **R'** defined as follow

(25) $$\mathbf{R'} = \mathbf{R}(\alpha, \beta, \gamma)\mathbf{Q}(\varphi, \theta).$$

$R(\alpha,\beta,\gamma)$ is given by Eq. (6) and $Q(\varphi,\theta)$ is given by (reminding Eqs.(5))

(26) $$\mathbf{Q}(\varphi, \theta) = \mathbf{R}_{\mathbf{z}}(\varphi)\mathbf{R}_{\mathbf{y}}(\theta)\mathbf{R}_{\mathbf{z}}(-\varphi).$$

The x_V', y_V' and z_V' parameters are given by the components of the following vector

(27) $$\mathbf{r}_{\mathbf{V}}' = q\mathbf{R}(\alpha, \beta, \gamma)[\mathbf{I} - \mathbf{Q}(\varphi, \theta)]\mathbf{n} + \mathbf{r}_{\mathbf{V}},$$

where $\mathbf{r}_V=(x_V, y_V, z_V)^T$ and $\mathbf{n}$ is the unit vector defined as

(28) $$\mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

The complete description of the process to obtain the above formulas is shown in Appendix 2.

Appendix 3 reports the relevant formulas for the on-sphere movement written in C code

Appendix I. Error estimation between true and implemented relative positioning

Comparing Eq. (12) and (15), the error $\boldsymbol{\delta}\mathbf{r}_V$ in the vertex V positioning is given by

(29) $$\boldsymbol{\delta}\mathbf{r}_V = \left[\mathbf{I} - \mathbf{R}(\Delta\alpha_0, \Delta\beta_0, \Delta\gamma_0)\right]\mathbf{r}_{V0}.$$

The range of α, β and γ is ±0.25deg and the range of vertex displacement is ±5mm. The worse error is given when the hexapod is positioned to one extreme of the range and a full range delta-command is issued. In this case the error in vertex location is not negligible and is approximately 0.09 mm.

Comparing Eq. (12) and (15), the errors $\delta\alpha$, $\delta\beta$, $\delta\gamma$ in the rotation angles are given by solution of the following equation

(30) $$\mathbf{R}(\Delta\alpha_0, \Delta\beta_0, \Delta\gamma_0)\mathbf{R}(\alpha_0, \beta_0, \gamma_0) = \mathbf{R}(\alpha_0 + \Delta\alpha_0 - \delta\alpha, \beta_0 + \Delta\beta_0 - \delta\beta, \gamma_0 + \Delta\gamma_0 - \delta\gamma)$$

Letting $\mathbf{R}_0$ the left-hand side matrix of previous equation and using Eqs. (7)-(9) we have

$$\Delta\alpha_0 - \delta\alpha = \arctan\frac{R_{0,32}}{R_{0,33}} - \alpha_0$$

(31) $$\Delta\beta_0 - \delta\beta = -\arcsin R_{0,32} - \beta_0$$

$$\Delta\gamma_0 - \delta\gamma = \arctan\frac{R_{0,21}}{R_{0,11}} - \gamma_0$$

The range of α, β and γ is ±0.25deg. The worse error is given when the hexapod is positioned to one extreme of the range (0.25deg) and a SET_POS_REL is issued with the full range (-0.5deg).In this extreme case the error is not negligible and is approximately 8arcsec. This error can be overcome issuing a SET_POS_REL command with angular parameters given by the right-side expressions of previous equations instead of $\Delta\alpha_0$, $\Delta\beta_0$ and $\Delta\gamma_0$.

Appendix II. Detailed formulation of on-sphere positioning

In the spherical positioning the shell of M2 has to rotate around a point that is placed on the actual optical axis of the shell itself as shown in Fig. 3. The RoC q is positive when the rotation point is toward M1.
Let's define as α_0, β_0, γ_0 and $\mathbf{r}_{V0}=(x_{V0}, y_{V0}, z_{V0})^T$ as parameters stating the current position in the MRS. As shown in Fig. 2 let's define as $V\xi\eta\zeta$ the Shell Coordinate System (SCS) that is fixed with the shell. Inverting Eq (3) we obtain the transform giving the coordinates $\boldsymbol{\rho}=(\xi,\eta,\zeta)$ in the SCS from the MRS coordinates $\mathbf{r}$ as follow

(32) $$\boldsymbol{\rho} = \mathbf{R}^T(\alpha, \beta, \gamma)(\mathbf{r} - \mathbf{r}_V).$$

The direction of rotation is identified by two angles φ and θ as shown in Fig. 3. The angle φ defines the direction from the ξ axis of the movement. The angle θ defines the amount of rotation from the optical axis.

The target position is reached by the following sequence of transformations

- displacement of vertex V to the center of rotation C at distance q from V in direction $-\zeta$ (obtaining the $C\xi_1\eta_1\zeta_1$ coordinate system);
- rotation of the new coordinate system of an angle φ around ζ_1 (obtaining the $C\xi_2\eta_2\zeta_2$ coordinate system);
- rotation of the new coordinate system of an angle θ around η_2 (obtaining the $C\xi_3\eta_3\zeta_3$ coordinate system);
- back-rotation of the new coordinate system of an angle -φ around ζ_3 (obtaining the $C\xi_4\eta_4\zeta_4$ coordinate system);
- displacement of the new coordinate system toward the z4 axis for a distance q (obtaining the final $V'\xi'\eta'\zeta'$ coordinate system).

The coordinates in the final system are given by

(33) $\boldsymbol{\rho}' = \mathbf{R}_z^T(-\varphi)\mathbf{R}_y^T(\theta)\mathbf{R}_z^T(\varphi)(\boldsymbol{\rho} - q\mathbf{n}) + q\mathbf{n}$

where

(34) $\mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$

Inverting Eq. (33) we obtain

(35) $\boldsymbol{\rho} = \mathbf{R}_z(\varphi)\mathbf{R}_y(\theta)\mathbf{R}_z(-\varphi)(\boldsymbol{\rho}' - q\mathbf{n}) + q\mathbf{n}$

or better

(36) $\boldsymbol{\rho} = \mathbf{Q}(\varphi,\theta)\boldsymbol{\rho}' + q[I - \mathbf{Q}(\varphi,\theta)]\mathbf{n}$

where

(37) $\mathbf{Q}(\varphi,\theta) = \mathbf{R}_z(\varphi)\mathbf{R}_y(\theta)\mathbf{R}_z(-\varphi)$

Reminding Eq. (3)

(38) $\mathbf{r}' = \mathbf{R}(\alpha,\beta,\gamma)\mathbf{r} + \mathbf{r}_V$

we can define in a similar way the target position in the MRS to be

(39) $\mathbf{r}'' = \mathbf{R}'\mathbf{r} + \mathbf{r}_V{}'$,

where

(40) $\mathbf{R}' = \mathbf{R}(\alpha',\beta',\gamma')$

Because of the "rigid" (no-deforming) transformations from MRS to V'$\xi'\eta'\zeta'$ and V$\xi\eta\zeta$, $\boldsymbol{\rho}'=\mathbf{r}$ in Eq. (36) and $\mathbf{r}=\boldsymbol{\rho}$ in Eq. (3). Combining the previous equations with the above assumptions we have

(41) $\mathbf{R}(\alpha',\beta',\gamma')\mathbf{r} + \mathbf{r}_V{}' = \mathbf{R}(\alpha,\beta,\gamma)\mathbf{Q}(\varphi,\theta)\mathbf{r} + q\mathbf{R}(\alpha,\beta,\gamma)[\mathbf{I} - \mathbf{Q}(\varphi,\theta)]\mathbf{n} + \mathbf{r}_V$

in particular

(42) $\mathbf{r}_V{}' = q\mathbf{R}(\alpha,\beta,\gamma)[\mathbf{I} - \mathbf{Q}(\varphi,\theta)]\mathbf{n} + \mathbf{r}_V$

and

(43) $\mathbf{R}' = \mathbf{R}(\alpha',\beta',\gamma') = \mathbf{R}(\alpha,\beta,\gamma)\mathbf{Q}(\varphi,\theta)$

From **R'** components the rotation angles α', β' and γ' can be obtained (see Eqs. (7)-(9))

(44) $\alpha' = \arctan(R'_{32}/R'_{33})$

(45) $\beta' = -\arcsin(R'_{31})$

(46) $\gamma' = \arctan(R'_{22}/R'_{11})$

At the end, Eqs. (22)-(24) and (27) give the complete set of parameters α', β', γ', x_V', y_V' and z_V' for the SET_POS_ABS command starting from the α', β', γ', x_V', y_V' and z_V' parameters identifying the current global position (read-back from the hexapod control electronics) and the user defined q, φ and θ parameters.

Appendix III. True-Relative positioning formulas in C format

In order to simplify the software writing the explicit expression of relevant quantities in C format are given in the following. Angles α, β, γ are identified with `a`, `b` and `c` letters, Δ with `d` and the prime sign with `1` suffix.

Template of call: true_relative_positioning(da,db,dc,dxv,dxy,dxz)

(Current absolute position is supposed to be read inside the call or previously set in global variable)

Letting `a`, `b`, `c`, `xv`, `yv` and `zv` the current absolute parameters defining the hexapod positioning and `da`, `db`, `dc`, `dxv`, `dyv` and `dzv` the target parameters of the true relative hexapod movement command, the SET_ABS_POS parameter `a1`, `b1`, `c1`, `xv1` `yv1` and `zv1` to perform the target relative movement are:

```
a1 = atan(R1_32/R1_33)

b1 = -asin(R1_31)

c1 = atan(R1_21/R1_11)

xv1 = dxv + xv*Cos(db)*Cos(dc) + Cos(dc)*(zv*Cos(da)
    + yv*Sin(da))*Sin(db)+(-(yv*Cos(da)) + zv*Sin(da))*Sin(dc)

yv1 = dyv + yv*Cos(da)*Cos(dc) - zv*Cos(dc)*Sin(da)
    + (xv*Cos(db) + (zv*Cos(da) + yv*Sin(da))*Sin(db))*Sin(dc)

zv1 = dzv + zv*Cos(da)*Cos(db) + yv*Cos(db)*Sin(da) - xv*Sin(db)
```

The The explicit expression of the relevant elements of the $\mathbf{R}'$ matrix are given by:

```
R1_32 = Cos(b)*Cos(db)*Sin(a)*Sin(da)

R1_33 = Cos(a)*Cos(b)*Cos(da)*Cos(db)

R1_31 = Sin(b)*Sin(db)

R1_21 = Cos(b)*Cos(db)*Sin(c)*Sin(dc)

R1_11 = Cos(b)*Cos(c)*Cos(db)*Cos(dc)
```

Appendix IV. Move-on-sphere positioning formulas in C format

In order to simplify the software writing the explicit expression of relevant quantities in C format are given in the following. Angles α, β and γ are identified with `a`, `b` and `c` letters, the prime sign with 1 suffix.

Template of call: spherical_positioning(theta,phi,q)

(Current absolute position is supposed to be read inside the call or previously set in global variable)

Letting `a`, `b`, `c`, `xv`, `yv` and `zv` the current absolute parameters defining the hexapod positioning and `phi`, `theta` and `q` the target parameters of the spherical movement (movement direction on the sphere, angular displacement and RoC), the SET_ABS_POS parameter `a1`, `b1`, `c1`, `xv1` `yv1` and `zv1` to perform the target relative movement are:

```
a1 = atan(R1_32/R1_33)

b1 = -asin(R1_31)
```

```
c1 = atan(R1_21/R1_11)

x1V = xv + q*((-1 + Cos(theta))*(Cos(a)*Cos(c)*Sin(b) + Sin(a)*Sin(c))
    + Cos(b)*Cos(c)*Cos(phi)*Sin(theta) + (Cos(c)*Sin(a)*Sin(b)
    - Cos(a)*Sin(c))*Sin(theta)*Sin(phi))

y1V = yv + q*Cos(c)*(Sin(a) - Cos(theta)*Sin(a) + Cos(a)*Sin(theta)*Sin(phi))
    + q*Sin(c)*(Cos(a)*(-1 + Cos(theta))*Sin(b) + Sin(theta)*(Cos(b)*Cos(phi)
    + Sin(a)*Sin(b)*Sin(phi)))

z1V = zv + q*(Cos(a)*Cos(b)*(-1 + Cos(theta)) + Sin(theta)*(-(Cos(phi)*Sin(b))
    + Cos(b)*Sin(a)*Sin(phi)))
```

The explicit expression of the relevant elements of the $\mathbf{R'}$ matrix are given by:

```
R1_32 = -((-1 + Cos(theta))*Cos(phi)*Sin(b)*Sin(phi))
      + Cos(b)*(-(Cos(a)*Sin(theta)*Sin(phi)) + Sin(a)*(Power(Cos(phi),2)
      + Cos(theta)*Power(Sin(phi),2)))

R1_33 = Cos(a)*Cos(b)*Cos(theta) + Sin(theta)*(-(Cos(phi)*Sin(b))
      + Cos(b)*Sin(a)*Sin(phi))

R1_31 = -(Cos(a)*Cos(b)*Cos(phi)*Sin(theta))
      - 2*Cos(b)*Cos(phi)*Sin(a)*Power(Sin(theta/2),2)*Sin(phi)
      - Sin(b)*(Cos(theta)*Power(Cos(phi),2) + Power(Sin(phi),2))

R1_21 = Cos(phi)*((Cos(c)*Sin(a)-Cos(a)*Sin(b)*Sin(c))*Sin(theta)
      + (Cos(theta)-1)*(Cos(a)*Cos(c)+Sin(a)*Sin(b)*Sin(c))*Sin(phi))
      + Cos(b)*Sin(c)*(Cos(theta)*Power(Cos(phi),2)+ Power(Sin(phi),2))

R1_11 = Cos(phi)*(-((Cos(a)*Cos(c)*Sin(b) + Sin(a)*Sin(c))*Sin(theta))
      + (-1 + Cos(theta))*(Cos(c)*Sin(a)*Sin(b) - Cos(a)*Sin(c))*Sin(phi))
      + Cos(b)*Cos(c)*(Cos(theta)*Power(Cos(phi),2) + Power(Sin(phi),2))
```

References

[1] ADS International srl, "*M2 Hexapod Software Protocol*", CAN# 616a046, Issue C, 04 Nov 2005

Doc_info_start
Title: Algorithm defining hexapod parameters for rotating M2 around an arbitrary M2 on-axis point and performing a true relative roto-translation movement
Document Type: Technical Report
Source: INAF – Osservatorio Astrofisico di Arcetri
Issued by: A. Riccardi
Date_of_Issue:24 Nov 2008
Revised by:
Date_of_Revision:
Checked by:
Date_of_Check:
Accepted by:
Date_of_Acceptance:
Released by:
Date_of_Release:
File Type: MS-WORD
Local Name:616f001E
Category:Telescope Auxiliaries
Sub-Category:M2 F/15
Assembly:M2 F/15 General
Sub-Assembly:M2 Hexapod software protocol
Part Name:
CAN designation: 616f001
Revision: E
Doc_info_end