Aristotle University of Thessaloniki
Department of Electrical and Computer Engineering
Electronics Division

Αθανάσιος Κωνσταντής (10537)

Ρομποτική

Εργασία 2024

1 Ερώτημα 1

Θέλουμε να εκφράσουμε τη θέση και την ταχύτητα του αγκώνα $\boldsymbol{p}_e(t)$ και $\dot{\boldsymbol{p}}_e(t)$ συναρτήσει της γενικευμένης θέσης και ταχύτητας του καρπού $(\boldsymbol{p}_w(t), \boldsymbol{Q}_w(t), \dot{\boldsymbol{p}}_w(t), \boldsymbol{\omega}_w(t))$.

Είναι εύκολο να βρούμε τον πίνακα ομογενούς μετασχηματισμού από το πλαίσιο $\{\mathbf{w}\}$ στο πλαίσιο $\{\mathbf{e}\}$. Έστω $\boldsymbol{R}_w$ ο πίνακας στροφής του $\boldsymbol{Q}_w$ quaternion. Βλέποντας το Σχήμα 2 της εκφώνησης, ισχύει:

$$\boldsymbol{p}_e = \boldsymbol{p}_w + l_f * (-\boldsymbol{n}_w)$$
$$\boldsymbol{R}_e = \boldsymbol{R}_w \cdot Rot(\boldsymbol{o}_w, \theta)$$

Η πρώτη εξίσωση είναι λογικό από πού προκύπτει. Η δεύτερη, προκύπτει από την παρατήρηση ότι το πλαίσιο $\{\mathbf{e}\}$ είναι περιστραμένο σε σχέση με το $\{\mathbf{w}\}$ κατά γωνία θ ως προς τον άξονα $\boldsymbol{o}_w$. Συνεπώς πρέπει να πολλαπλασιάσουμε με έναν πίνακα περιστροφής κατά γωνία θ στον άξονα $\boldsymbol{o}_w$ από τα δεξιά και όχι από τα αριστερά, επειδή γνωρίζουμε τον άξονα αυτόν ως τον άξονα $\boldsymbol{y}$ του πλαισίου $\{\mathbf{w}\}$.

Για να βρούμε τη γωνία θ, αρκεί να βρούμε το εσωτερικό γινόμενο:

$$\boldsymbol{p}_e \cdot d\boldsymbol{p}, \quad d\boldsymbol{p} = \boldsymbol{p}_w - \boldsymbol{p}_e$$

Τελικά έχουμε:

$$\cos(\theta) = -\frac{\boldsymbol{p}_e \cdot d\boldsymbol{p}}{|\boldsymbol{p}_e| \cdot |d\boldsymbol{p}|}$$
$$\theta = -\arccos\left(\frac{\boldsymbol{p}_e \cdot d\boldsymbol{p}}{|\boldsymbol{p}_e| \cdot |d\boldsymbol{p}|}\right)$$

Οπότε ισχύουν:

$$\boldsymbol{p}_e(t) = \boldsymbol{p}_w(t) + \boldsymbol{R}_w(t) \cdot \begin{bmatrix} -l_f \\ 0 \\ 0 \end{bmatrix}$$
$$\dot{\boldsymbol{p}}_e(t) = \dot{\boldsymbol{p}}_w(t) + \dot{\boldsymbol{R}}_w(t) \cdot \begin{bmatrix} -l_f \\ 0 \\ 0 \end{bmatrix} = \dot{\boldsymbol{p}}_w + \hat{\boldsymbol{\omega}}_w(t)\boldsymbol{R}_w(t) \begin{bmatrix} -l_f \\ 0 \\ 0 \end{bmatrix}$$

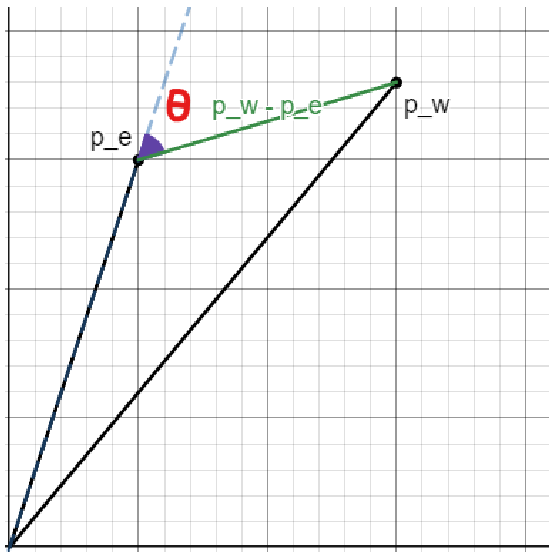

Figure 1: Παραπάνω φαίνεται το διάγραμμα, με τη γωνία $|\vartheta|$, τα διανύσματα $\boldsymbol{p}_e$, $\boldsymbol{p}_w$ και το διάνυσμα $d\boldsymbol{p}$

2 Ερώτημα 2

Θέλουμε να σχεδιάσουμε κατάλληλο νόμο ελέγχου ώστε το ρομπότ να εκτελεί την ένδυση του χεριού σε χρόνο $T = 5\text{sec}$ ανά τμήμα και περίοδο δειγματοληψίας $T_s = 0.01\text{sec}$.

Αρχικά θα πρέπει να βρούμε την κατάλληλη τροχιά του βραχίονα ώστε να γίνει η ένδυση σωστά. Θα συμπεριφερθούμε στην θέση και στον προσανατολισμό σαν να είναι δυο διαφορετικά αποσυζευγμένα μεγέθη τα οποία στο τέλος θα ενεργήσουν μαζί ώστε να δημιουργήσουν την τροχιά. Έστω $\boldsymbol{G}(t)$ ο ομογενής μετασχηματισμός του άκρου του βραχίονα στον χρόνο. Αρχικά, ξεκινάμε έχοντας:

$$\boldsymbol{G}(0) = \begin{bmatrix} \boldsymbol{R}_w(0) & \boldsymbol{p}_w(0) \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Είναι σημαντικό να ειπωθεί ότι παρατηρήθηκε κάποια διαφορά στον προσανατολισμό του πλαισίου του ρομπότ στην αρχική θέση σε σχέση με αυτόν του ανθρώπινου χεριού. Αυτό οφείλεται πιθανόν σε κάποιο toolbox του Matlab το οποίο τοπικά στον υπολογιστή μας έπαιρνε διαφορετικά αποτελέσματα του $\boldsymbol{Q}_w$ από την **get_arm_posture()**. Δοκιμάστηκε να τρέξουμε τον ίδιο ακριβώς κώδικα στο Matlab Online όπου έβγαιναν τα σωστά αποτελέσματα.

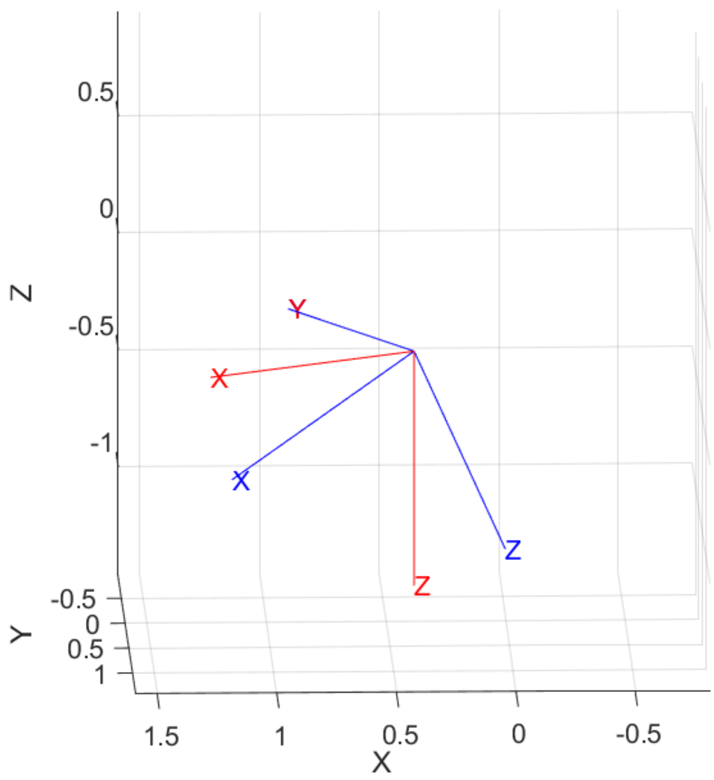

Figure 2: Με μπλε βλέπουμε το πλαίσιο που βγαίνει ως output από την get_human_arm και με κόκκινο είναι το πλαίσιο που θα έπρεπε να βγαίνει ως output, το οποίο υπολογίζουμε μέσω του ευθύ κινηματικού στην αρχική θέση (και επίσης το βλέπουμε και από το Matlab Online ως output της get_human_arm). Παρατηρούμε ότι υπάρχει μια στροφή κατά τον y άξονα περίπου 1 rad

Τελικά καταλήξαμε σε μια λύση που δίνει σωστά αποτελέσματα και στις δύο περιπτώσεις. Έστω $\boldsymbol{g}_{06}$ ο πίνακας ομογενούς μετασχηματισμού του άκρου το βραχίονα στην αρχή της κίνησης (t=0). Υπολογίζουμε το:

$$\boldsymbol{g}_{6w} = \boldsymbol{g}_{06}^{-1}\boldsymbol{g}_{0w};$$

Υπό κανονικές συνθήκες, ο πίνακας $\boldsymbol{g}_{6w} = \boldsymbol{I}_4$ αν το αποτέλεσμα που βγάζει η συνάρτηση **get_arm_posture()** είναι σωστό. Αλλιώς, ο πίνακας $\boldsymbol{g}_{6w}$ περιγράφει μόνο μια στροφή του πλαισίου ως προς τον άξονα y. Με αυτόν πολλαπλασιάζεται κάθε μετάβαση από το $\{\mathbf{0}\}$ στο $\{\mathbf{6}\}$ για να διορθώσουμε το λανθασμένο αποτέλεσμα. Αυτή τη διόρθωση την κάναμε επειδή δεν μπορέσαμε να βρούμε τι ήταν αυτό που δημιουργούσε το λανθασμένο output της get_arm_posture() μιας και δεν μπορούμε να δούμε τον κώδικά της. Τα versions των toolboxes ήταν σωστά, ο κώδικας ήταν ίδιος, οπότε παραμένει μυστήριο το ποιό ήταν το πρόβλημα*.

Ας μελετήσουμε τη θέση του άκρου του βραχίονα στον χρόνο. Θα θέλαμε για τα πρώτα 0-5 sec να ξεκινήσει από το σημείο $\boldsymbol{p}_w$ και να καταλήγει στο σημείο $\boldsymbol{p}_e$. Επιλέγουμε να κάνουμε την κίνηση αυτή γραμμικά, άρα καταλήγουμε στην εξίσωση:

$$\boldsymbol{d}(t) = \boldsymbol{p}_w + 0.2t(\boldsymbol{p}_e - \boldsymbol{p}_w), \quad 0 \leq t \leq 5sec \tag{1}$$

Με παρόμοιο τρόπο σχεδιάζουμε την κίνηση για χρόνο από 5-10 sec όπου θέλουμε ο βραχίονας να ξεκινάει από το σημείο $\boldsymbol{p}_e$ και να καταλήγει στο (0,0,0).

$$\boldsymbol{d}(t) = \boldsymbol{p}_e - 0.2(t-5)\boldsymbol{p}_e, \quad 5 \leq t \leq 10sec \tag{2}$$

Αντικαθιστώντας στις (1), (2) τον τύπο $\boldsymbol{p}_e(t) = \boldsymbol{p}_w(t) + \boldsymbol{R}_w(t) \cdot \begin{bmatrix}-l_f & 0 & 0\end{bmatrix}^T$ μπορούμε να υπολογίσουμε την θέση του άκρου του βραχίονα στον χρόνο.

Για τον προσανατολισμό, έστω $\boldsymbol{R}_d(t)$ ο πίνακας στροφής στον χρόνο. Έχουμε:

$$\boldsymbol{R}_d(t) = \boldsymbol{R}_w(t) \quad 0 \leq t \leq 3 \tag{3}$$

Επιλέγουμε από την χρονική στιγμή t = 3 sec εώς t = 7 sec, να μεταβούμε από τον προσανατολισμό του πήχη στον προσανατολισμό του μπράτσου με γραμμικό τρόπο ώστε να μην έχουμε απότομες κινήσεις του ρομπότ στον χρόνο. Ο υπολογισμός μιας τέτοιας μετάβασης δεν είναι τόσο απλός όσο αυτός που κάναμε για τη θέση, και το toolbox που χρησιμοποιούμε μας έχει έτοιμη συνάρτηση για να το καταφέρουμε. Χρησιμοποιούμε λοιπόν τη συνάρτηση **quatinterp()**, η οποία μας δίνει τη γραμμική παρεμβολή από ένα quaternion σε ένα άλλο.

Το αρχικό quaternion είναι γνωστό και είναι το $\boldsymbol{Q}_w$. Το τελικό quaternion είναι αυτό που περιγράφεται από τον πίνακα $\boldsymbol{R}_e = \boldsymbol{R}_w \cdot Rot(\boldsymbol{o}_w, \theta)$, έστω $\boldsymbol{Q}_e$.

Τέλος, ο προσανατολισμός για χρόνο 7-10 sec είναι:

$$\boldsymbol{R}_d(t) = \boldsymbol{R}_e(t) \tag{4}$$

Έχοντας ορίσει τη θέση και τον προσανατολισμό του ρομπότ για κάθε χρονική στιγμή, μπορούμε εύκολα να ορίσουμε τον πίνακα ομογενούς μετασχηματισμού του άκρου στον χρόνο:

$$\boldsymbol{G}(t) = \begin{bmatrix} \boldsymbol{R}_d(t) & \boldsymbol{d}(t) \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Για τον υπολογισμό της ταχύτητας που πρέπει να έχει το άκρο για να ακολουθήσει πιστά την τροχιά μας, θα χρησιμοποιήσουμε την προσέγγιση της παραγώγου Euler. Έστω $\boldsymbol{d}(k) = \boldsymbol{d}(kT_s)$ το σήμα δειγματοληψίας της τροχιάς μας την χρονική στιγμή kT_s.

$$\dot{\boldsymbol{p}}_d(k) \approx \frac{\boldsymbol{d}(k+1) - \boldsymbol{d}(k)}{T_s}$$

$$\dot{\boldsymbol{R}}_d(k) \approx \frac{\boldsymbol{R}_d(k+1) - \boldsymbol{R}_d(k)}{T_s}$$

Με αυτόν τον τρόπο μπορούμε να υπολογίσουμε το $\boldsymbol{\omega}_d$:

$$\hat{\boldsymbol{\omega}}_d(k) = \dot{\boldsymbol{R}}_d(k)\boldsymbol{R}_d(k)^T$$

Από τον τύπο των αντισυμμετρικών πινάκων υπολογίζουμε το $\boldsymbol{\omega}_d$ από το $\hat{\boldsymbol{\omega}}_d$

Θα υλοποιήσουμε κινηματικό έλεγχο του άκρου του βραχίονα. Υπολογίζουμε πρώτα το σφάλμα

$$\boldsymbol{e}_p = \boldsymbol{p}(k) - \boldsymbol{p}_d(k+1)$$

Όπου $\boldsymbol{p}$ είναι η πραγματική θέση του άκρου και $\boldsymbol{p}_d$ είναι η θέση που θέλουμε να έχει το άκρο την επόμενη χρονική στιγμή δειγματοληψίας.

Επίσης, υπολογίζουμε το quaternion σφάλματος προσανατολισμού:

$$\boldsymbol{Q}_{error} = \boldsymbol{Q}_r(k) * \boldsymbol{Q}_d^{-1}(k+1)$$

όπου $\boldsymbol{Q}_r$ είναι ο πραγματικός προσανατολισμός του άκρου και $\boldsymbol{Q}_d$ είναι ο προσανατολισμός που θέλουμε να έχει το άκρο την επόμενη χρονική στιγμή δειγματοληψίας.

Ορίζουμε το σφάλμα:

$$\boldsymbol{e}_o = \sin\left(\frac{\theta_e}{2}\right)\boldsymbol{k_e}$$

που είναι τα τελευταία 3 στοιχεία του $\boldsymbol{Q}_{error}$. Τέλος, υλοποιούμε τον κινηματικό έλεγχο στο σύστημα του άκρου $\boldsymbol{V} = \boldsymbol{u}$, όπου $\boldsymbol{V} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix}$ και $\boldsymbol{u}$ η είσοδος ελέγχου:

$$\boldsymbol{u} = \begin{bmatrix} \dot{\boldsymbol{p}}_d \\ \boldsymbol{\omega}_d \end{bmatrix} - \boldsymbol{K}\begin{bmatrix} \boldsymbol{e}_p \\ \boldsymbol{e}_o \end{bmatrix}$$

και επιλέγουμε $\boldsymbol{K} = \boldsymbol{I}_6$, Άρα, έχουμε την ταχύτητα που έχει το άκρο τη συγκεκριμένη χρονική στιγμή.

Έστω $\boldsymbol{J}(k)$ η Ιακωβιανή του άκρου στον χρόνο, την οποία μπορούμε να υπολογίσουμε μέσω της μεθόδου **robot.jacob0(q))**, όπου **q**(k) είναι το διάνυσμα των θέσεων των αρθρώσεων του ρομπότ την χρονική στιγμή kT_s. Τότε ισχύει:

$$\dot{\boldsymbol{q}}(k) = \boldsymbol{J}^{-1}(k)\boldsymbol{V}(k)$$

και τελικά μέσω της προσέγγισης Euler μπορούμε να υπολογίσουμε τις νέες θέσεις $\boldsymbol{q}(k)$ ως:

$$\boldsymbol{q}(k+1) = \boldsymbol{q}(k) + \dot{\boldsymbol{q}}(k)T_s$$

Η Ιακωβιανή είναι πάντα αντιστρέψιμη επειδή πρόκειται για τετραγωνικό πίνακα 6x6 και δεν περνάμε από κανένα ιδιάζον σημείο. Συνεπώς, έχουμε προσεγγιστική λύση για κάθε σημείο της τροχιάς μας.

Τρέχουμε την προσομοίωση και καταλήγουμε με τα παρακάτω αποτελέσματα:

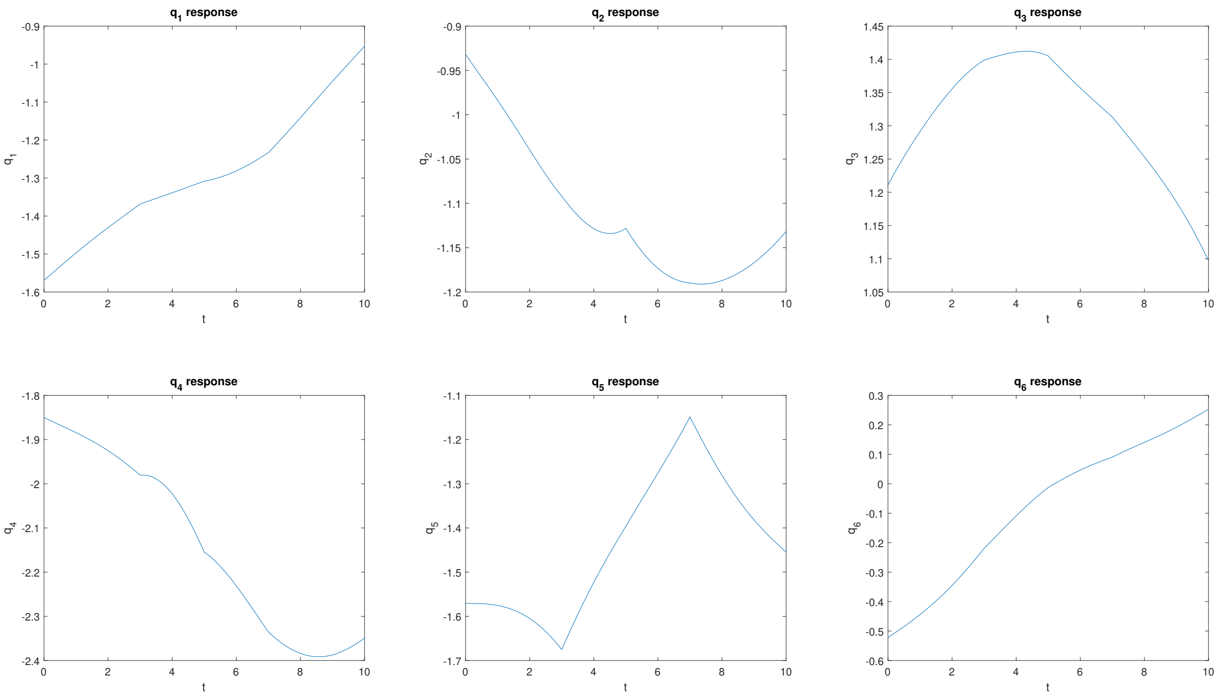

Figure 3: Απόκριση των μεταβλητών των αρθρώσεων στον χρόνο από την προσομοίωσή μας.

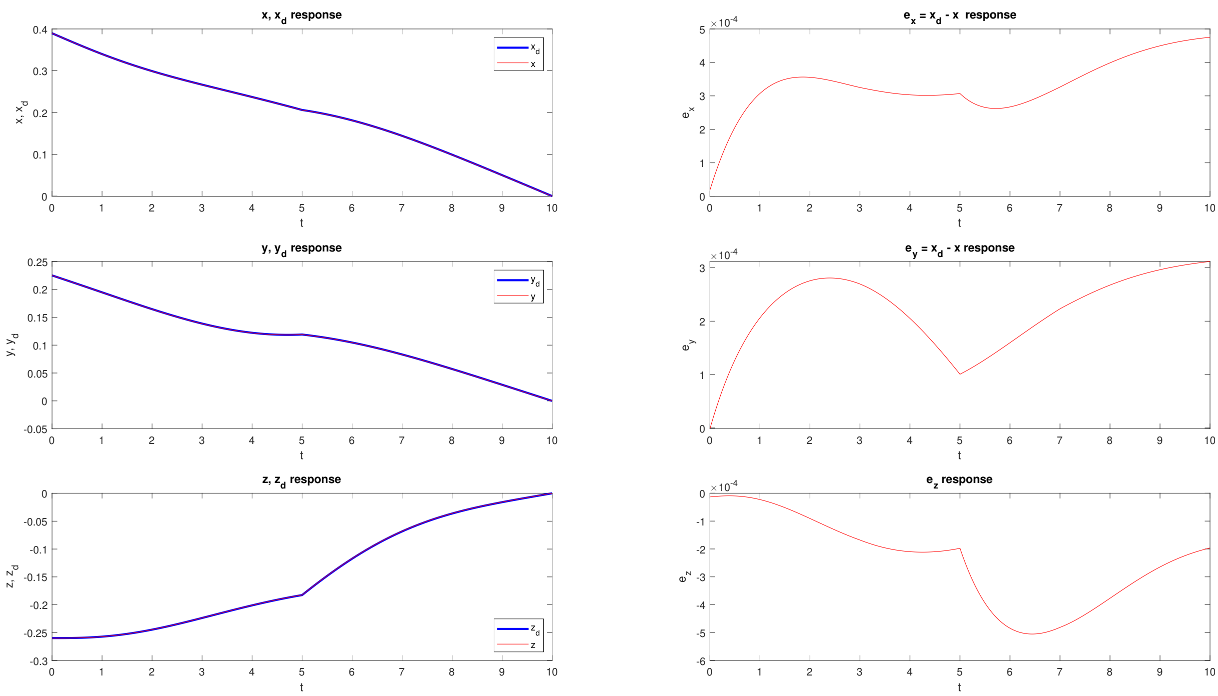

Figure 4: Απόκριση της θέσης του άκρου του ρομποτικού βραχίονα σε καρτεσιανές συντεταγμένες. Με μπλε βλέπουμε την απαιτούμενη τροχιά και με κόκκινο την πραγματική. Φαίνεται να συμπίπτουν, οπότε δεξιά δείχνουμε το σφάλμα, το οποίο είναι της τάξης του 10^{-4}

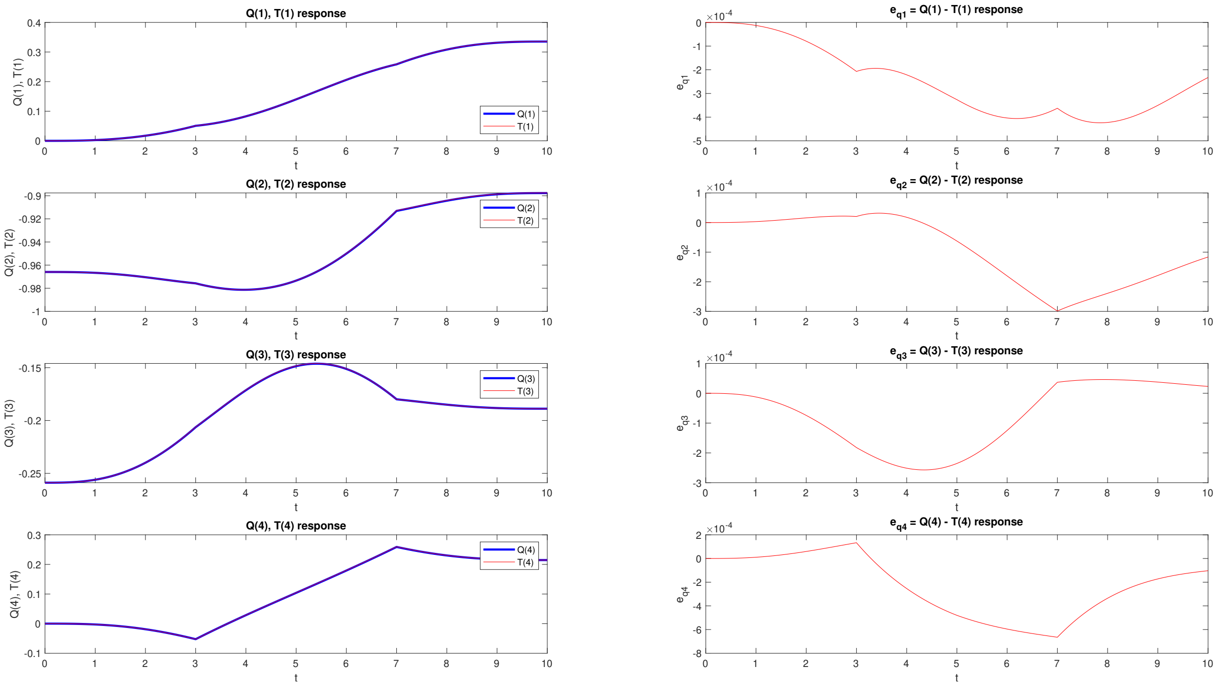

Figure 5: Απόκριση της περιστροφής του άκρου του ρομποτικού βραχίονα μέσω απεικόνισής της σε μοναδιαίο quaternion. Με μπλε βλέπουμε την απαιτούμενη περιστροφή και με κόκκινο την πραγματική. Στα δεξιά βλέπουμε τα σφάλματα μεταξύ των δύο αυτών μεγεθών.

Συμπέρασμα: Τα σφάλματά μας είναι της τάξης του 10^{-4} και θεωρούνται αρκετά μικρά ώστε να μπορούμε να πούμε ότι το μοντέλο μας είναι πολύ καλό. Αυτό φαίνεται και οπτικά βλέποντας το animation του ρομπότ μέσω του κώδικα που επισυνάπτουμε με την εργασία. Συνεπώς, καταφέραμε με επιτυχία τον στόχο να προσομοιώσουμε την ένδυση του ανθρώπινου χεριού. Φυσικά υπάρχουν πολλές λύσεις σε αυτό το πρόβλημα και ο κώδικάς μας είναι αρκετά εύκολο να τροποποιηθεί ώστε π.χ. η περιστροφή από τον πήχη στο μπράτσο να γίνει πιο γρήγορα, ή να χρησιμοποιηθεί η μέθοδος Runge-Kutta αντί για Euler για την λύση της διαφορικής. Παρ'όλα αυτά θα αρκεστούμε σε αυτή τη λύση μιας και το σφάλμα της είναι αρκετά μικρό ώστε να θεωρείται αποδεκτό.

**Το πρόβλημα με τον προσανατολισμό που αναφέρεται παραπάνω βρέθηκε και σε άλλους συμφοιτητές και ανέβηκαν καινούρια αρχεία. Αυτό συνέβη μετά την συγγραφή αυτού του report και άρα οι παραπάνω πληροφορίες είναι βασισμένες στα δεδομένα που υπήρχαν τότε. Ο κώδικάς μας θεωρητικά θα πρέπει να τρέχει σωστά και για τα δύο σετ δεδομένων σε οποιαδήποτε versions του toolbox.*