

Anthropomorphic mechanical design and Lyapunov-based control of a new shoulder rehabilitation system

Mahdieh Babaiasl^{a*}, Ahmad Ghanbari^{a,b}, and SMRS Noorani^a

^aSchool of Engineering Emerging Technologies, University of Tabriz, Tabriz, Iran

^bFaculty of Mechanical Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article history:

Received January 20, 2014

Received in Revised form

April, 10, 2014

Accepted 7 May 2014

Available online

9 May 2014

Keywords:

Anthropomorphic Mechanical Design

Lyapunov-based controller

Disturbance rejection

Robot-aided rehabilitation

Stroke

Tracking problem

Exoskeleton robots

Nonlinear control

ABSTRACT

Stroke is one of the main causes of disability. It affects millions of people worldwide. One symptom of stroke is disabled arm function. Restoration of arm function is necessary to resuming activities of daily living (ADL). Along with traditional rehabilitation techniques, robot-aided therapy has emerged recently. The control schemes of rehabilitation robots are designed for two reasons. First they are designed for passive rehabilitation in which the robot guides the patient's limb through a predefined path and second for active rehabilitation in which the patient initiates the movement and is partially assisted or resisted by the robotic device. This paper introduces a new robot for shoulder rehabilitation. The Shoulder Rehabilitation System (SRS) has three degrees of freedom (DOFs) for three rotational DOFs of the shoulder but additional translational DOFs of the shoulder are also allowed to avoid discomfort to the patient. A new open circular mechanism is proposed for the third joint that solves the known issues for rehabilitation robots such as long wiring and discomfort associated with closed mechanisms. Lyapunov-based controller with integral action is proposed to guide the robot through a predefined trajectory. Simulation results proved that the proposed controller can track the desired trajectory; reject constant bounded disturbance to the system and is robust due to its nonlinear nature. The proposed controller is designed to be used in passive rehabilitation.

1. Introduction

Stroke is one of the main causes of disability and loss of motor function particularly affecting older people. It affects more than one million people in European Union each year (Brainin et al., 2000; Thorvaldsen et al., 1995). In the United States more than 0.7 million people become affected by stroke each year (Lloyd-Jones et al., 2010). The result of stroke is partial destruction of cortical tissue which leads to impaired arm and hand motor function. According to Nakayama et al. (1994)

* Corresponding author.

E-mail addresses: m.babaiasl@gmail.com (M. Babaiasl)

only 18% of stroke survivors regain full motor function after six months. By considering the mentioned issues, using different therapy approaches is necessary to regain motor function and improve functional outcomes.

Optimal restoration of arm and hand function is essential to independently perform activities of daily living (ADLs). The most common approach in stroke rehabilitation is one-to-one manually-assisted training or physiotherapy. This approach is labor-intensive, time-consuming and expensive. Besides, training sessions are often shorter than required for an optimal therapeutic outcome, the therapy varies from one therapist to another and from one hospital to another and is based on theories and therapist's experience. Furthermore, manually-assisted training lacks repeatability and objective measures of patient performance and progress. Taking all these constraints into consideration, robots can help to improve rehabilitation and become an important tool in stroke rehabilitation. Robot-aided arm therapy is more intensive, of longer duration and more repetitive. Using robots, number and duration of training sessions can be increased, while reducing the number of therapists required per patient, which in turn yields to reduced personnel costs. Furthermore, robot-aided therapy provides quantitative measures and supports objective observation and evaluation of the rehabilitation progress. Several studies showed that robot-aided therapy indeed improves motor function more than conventional therapy (Prange et al., 2006; Fazekas et al., 2007; Kwakkel et al., 2008).

Because of the mentioned reasons, the use of robots in rehabilitation has been increased in recent years. Robots developed for rehabilitation can be categorized as end-effector based robots and exoskeleton type robots. End-effector based robots are connected to patient's hand or forearm at one point. Robot's axes generally do not correspond to the human-joint rotation axes. From mechanical point of view, end-effector based robots are easier to build and use. In rehabilitation, this class of robots cannot induce joint trajectories exactly matching the human joints. The advantageous features of these robots are that they can easily adjust to different arm lengths; they are simple, usable and cost-effective. The disadvantageous feature is that in general the arm posture or the individual joint interaction torques, are not fully determined by the robot because the patient and the robot interact just through one point i.e. the robot's end-effector. Exoskeleton type robots are body or wall-grounded and are attached at several locations along the limb. In order to induce exact joint trajectories and to match natural redundancy, the robot's joints must be aligned to coincide with the human joints. This feature is important because in the case of mismatch undesired reaction forces can be created in the human joints.

Lots of researchers around the world have developed some endeffector-based and exoskeleton type robots for rehabilitation of upper limb. MIT-MANUS (Krebs et al., 1998), Mirror Image Motion Enabler (MIME) (Lum et al., 2006), GENTLE/s (Loureiro et al., 2003), Bi-Manu-Track (Hesse et al., 2003), Assisted Rehabilitation and Measurement (ARM) guide (Reinkensmeyer et al., 2003) and REHAROB therapy system (Toth et al., 2003) are some of end-effector based robots. Dampace (Stienen et al., 2007), T-WREX (Sanchez et al., 2006), MGA-exoskeleton (Sanchez et al., 2003), L-EXOS (Frisoli et al., 2007) and ARMin I-III (Nef et al., 2007; Mihelj et al., 2007; Nef et al., 2009) are some of exoskeleton type robots. ARMin I can be stated as semi-exoskeleton robot since it has combined end-effector based structure with exoskeleton structure. Krebs et al. (1998), Hesse et al. (2003), Carignan et al. (2005), Frisoli et al. (2007) and Mihelj et al. (2007), implemented impedance control for active rehabilitation. Mihelj et al. (2007) implemented PD control and computed torque control (CTC) to passively guide the patient's arm through a predefined trajectory. Stienen et al. (2007) used controlled breaking on three rotational axis of the shoulder and one axis of the elbow, thus it is a passive exoskeleton. The control scheme used by Loureiro et al. (2003) is bead pathway. Rehabilitation robots are subjected to various disturbances such as patient limb tremor. Trajectory tracking errors are because of various disturbances such as measurement and modeling inaccuracies and load variances. For better trajectory tracking, the proposed control schemes should be capable of reducing the effects of uncertainties. These uncertainties can be considered as load torques on input signals. Some control algorithms such as inverse dynamics controller with integral action (Siciliano

& Sciavicco, 2009) can be used to reject disturbance input to the system. But the controller is highly sensitive to parameter uncertainties and relies on linearization and equation decoupling. Lyapunov-based controller is a type of passivity-based controllers which is not essentially dependent upon linearization or equation decoupling. The controller is robust due to its nonlinear nature and changing robot parameters do not have any influence in the performance of the controller. On the other hand it should also be considered that Lyapunov-based controller cannot easily be modified due to its structural complexity. This paper demonstrates a new robot for shoulder rehabilitation. Initially mechanical design and simulation of the robot in Solidworks is presented and kinematics and dynamics of the robot are derived. Afterwards a control algorithm is proposed and applied to the robot and finally simulation results are presented to verify the presented robotic system. The main advantages of this robot compared to similar ones are being light weight, its unique mechanism for third joint that solves the known issues for rehabilitation robots such as long wiring and discomfort associated with closed mechanisms, ease of use, more comfortable, also tracking performance of the controller and its robustness are guaranteed properly.

2. Mechanical design of the Robot

2.1. Mechanical design in Solidworks

After studying the properties of the upper limb of an adult person (Lee et al., 1983; Pons, J.L., 2008) such as mass, moments of inertia and lengths of different segments, an exoskeleton robot is designed for shoulder joint rehabilitation. The Shoulder Rehabilitation System (SRS) has three DOFs for shoulder flexion/extension, abduction/adduction and internal/external rotation. Fig. 1 demonstrates SRS system with a model of a human limb. The system can be wall or wheelchair-mounted. Fig. 2 depicts SRS detailed properties.

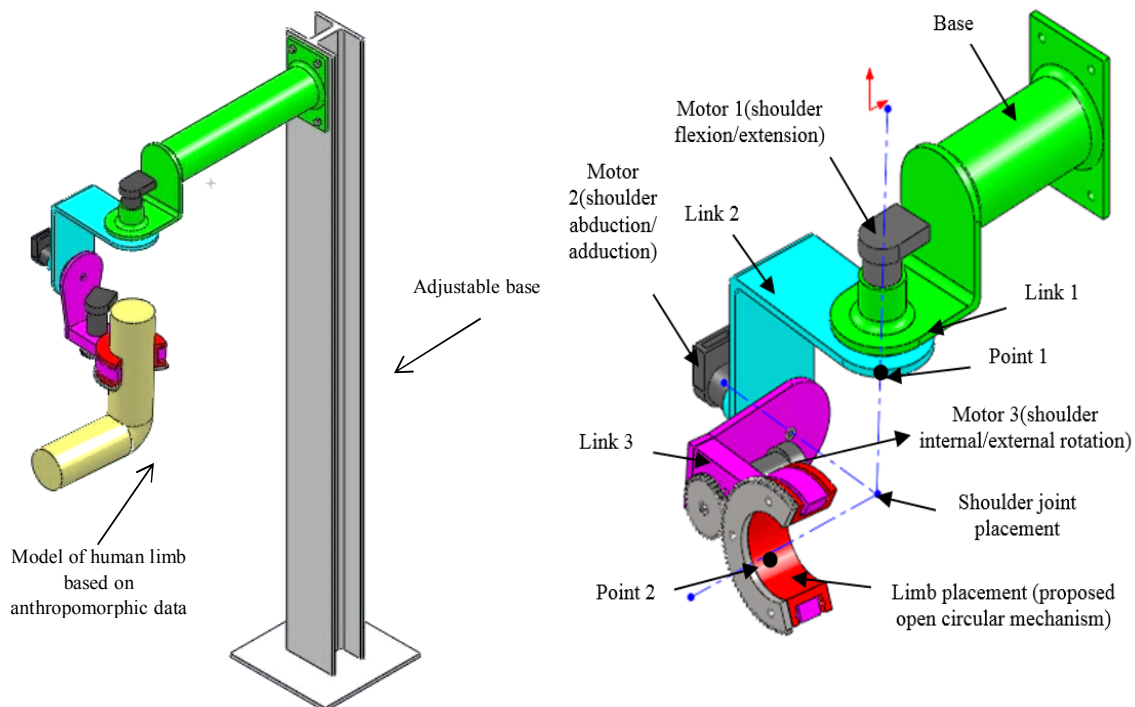


Fig. 1. Shoulder rehabilitation system with a model of a human limb

Fig. 2. Detailed properties of shoulder rehabilitation system

Shoulder flexion/extension is provided by motor 1. Link 1 holds motor 1 from one side and is fastened to the base from the other side. Link 2 which holds motor 2 is L-shaped in order for

comfortable accommodation of shoulder joint. Rotation axes of motor 1 and 2 intersect at a point which is shoulder joint placement point. For exact placement of the shoulder joint, a wheelchair with adjustable height is recommended. Human shoulder joint does not only have three rotational DOFs but it also possesses translational DOFs. In this robot translational movement of the shoulder joint is not a problem since that movement is allowed. Motor 2 provides shoulder abduction/adduction. In order to provide shoulder internal/external rotation, the rotation axis of motor 3 cannot be directly aligned with rotation axis of the limb due to anatomical configuration of the upper limb which causes discomfort to the patient. Most robots use gear mechanism with closed circular configuration but, this causes discomfort and pressure on the patient's limb. The alternative can be cable mechanism. But using cable mechanism has the drawback of losing connection because of long wiring. Considering these issues, a new open circular mechanism is proposed for this robot and power transmission is provided by the gear coupled to the shaft of motor 3. Fig. 3 depicts the mechanism of power transmission from motor in order to provide shoulder internal/external rotation. Fig. 4 depicts details of proposed open circular mechanism for joint 3.

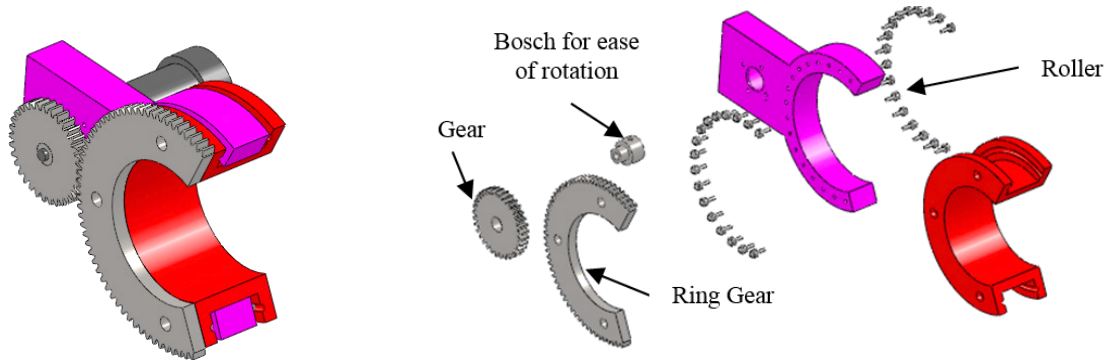


Fig. 3. mechanism of power transmission for joint 3 **Fig. 4.** Details of proposed circular mechanism

Another challenge in exoskeleton robots is that the robot should be adaptable to patient's limb in terms of segment lengths. This issue is not a problem here since the robot segments are of variable lengths. An important requirement for a rehabilitation robot is being light-weight. This feature lessens the loading effect on the patient limb. Aluminum is chosen as chassis material due to its low density and corrosion resistance. Actuators also affect the weight of the robot which most of the robot's weight is due to its actuators and they should be properly chosen according to maximum torque applied to each joint. The chosen actuators should be of light-weight, of low noise, of long life and of high power to weight ratio. The last feature is necessary for having a light weight and backdrivable robot.

2.2 Kinematics and Dynamics of the proposed robot

Direct kinematics of the robot is derived using Denavit-Hartenberg (DH) convention. Figure 5 depicts link frame assignment and Table 1 presents the DH parameters of the robot. a_i , d_i , α_i and θ_i are link length, link offset, link twist and joint angle respectively. Transformation matrices are given in appendix. The robot is in singular configuration when $\theta_2 = 0$ i.e. when the axis of rotation of the first joint is aligned with the axis of rotation of the third joint. Joint-space control algorithms do not need Jacobian or its inverse so singular points are not a problem. Cartesian space control algorithms need Jacobian or its inverse so singular points should be properly managed. For example, in these algorithms one can limit the motion of joint 2 to more than 10° to manage the singular points. In robotics, Jacobian is the mapping between end-effector velocity and joint velocity which is expressed as follows,

$$\mathbf{v}_e = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (1)$$

where, \mathbf{v}_e is a 6×1 vector including end-effector linear and angular velocity and $\dot{\mathbf{q}}$ is a 3×1 vector including joint velocities. \mathbf{J} , the Jacobian of the robot, is a 6×3 matrix. The Jacobian of the robot is defined as follows,

$$\mathbf{J} = \begin{pmatrix} \mathbf{z}_0 \times (\mathbf{P}_3 - \mathbf{P}_0) & \mathbf{z}_1 \times (\mathbf{P}_3 - \mathbf{P}_1) & \mathbf{z}_2 \times (\mathbf{P}_3 - \mathbf{P}_2) \\ \mathbf{z}_0 & \mathbf{z}_1 & \mathbf{z}_2 \end{pmatrix}. \quad (2)$$

In Eq. (2) \mathbf{z}_1 and \mathbf{z}_2 are the first three elements of the third column of \mathbf{T}_1^0 and \mathbf{T}_2^0 respectively. \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 are the first three elements of the last column of \mathbf{T}_1^0 , \mathbf{T}_2^0 and \mathbf{T}_3^0 respectively. These matrices are given in appendix. The Jacobian matrix of the robot is calculated as follows,

$$\mathbf{J} = \begin{pmatrix} -d_2 s_1 s_2 & d_2 c_1 c_2 & 0 \\ d_2 c_1 s_2 & d_2 s_1 c_2 & 0 \\ 0 & -d_2 s_2 & 0 \\ 0 & -s_1 & c_1 s_2 \\ 0 & c_1 & s_1 s_2 \\ 1 & 0 & c_2 \end{pmatrix}. \quad (3)$$

The Jacobian of this robot is first derived by hand and then is verified by Matlab. Deriving inverse kinematics of the robot is more difficult than direct kinematics since one should solve nonlinear equations; besides deriving an explicit solution is usually impossible and there are usually multiple or infinite solutions. Infinite solutions are in the case of redundant manipulators. If the Jacobian matrix is square, then using Eq. (1) one can obtain:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{v}}_e. \quad (4)$$

Then, joint angles are simply derived using integrating Eq. (4). But if Jacobian matrix is not square, pseudo-inverse is used to derive inverse kinematics and Eq. (4) can be expressed as:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{v}}_e, \quad (5)$$

where, \mathbf{J}^\dagger is right pseudo-inverse matrix of \mathbf{J} and is defined as follows:

$$\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1}. \quad (6)$$

Inverse kinematics of the robot is derived using Mathematica, since the software is great in symbolic math. Robot dynamics is derived using generalized d'Alembert method (Lee et al., 1983). This method gives more efficient equations compared to Lagrange and Newton Euler methods. Besides, computational burden are remarkably reduced which is ideal for control purposes.

Table 1. DH parameters of the robot

Joint i	a_i	d_i	α_i	θ_i
1	0	d_1	$-\pi/2$	θ_1
2	0	0	$\pi/2$	θ_2
3	0	d_2	0	θ_3

Robot dynamics is expressed as follows:

$$\boldsymbol{\tau} = \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \quad (7)$$

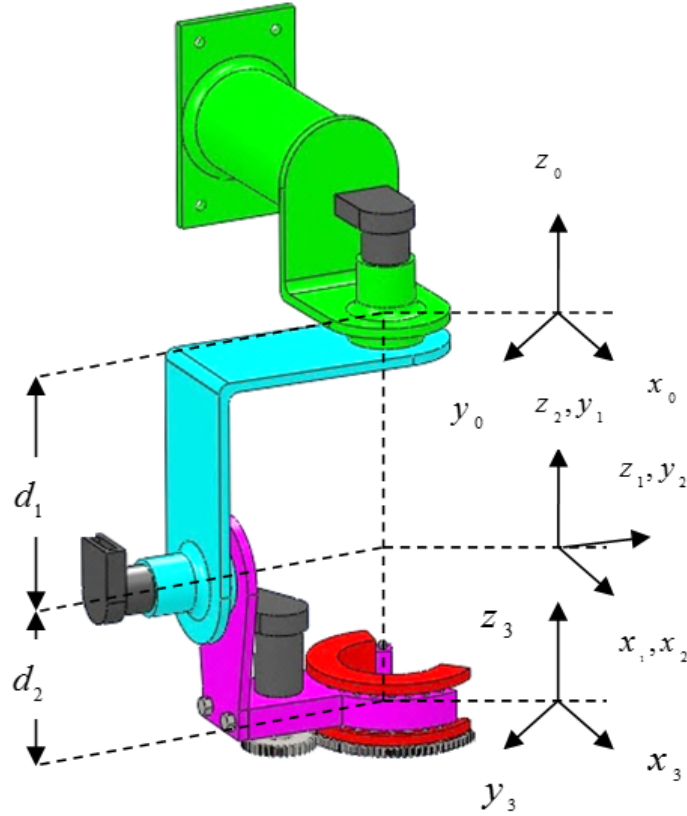


Fig. 5. Link frame assignments in DH convention

where, $D(q)$ is a 3×3 inertia matrix, $H(q, \dot{q})$ is a 3×1 vector of centrifugal and coriolis terms and $G(q)$ is a 3×1 vector of gravity term. These matrices are provided in appendix. τ is a 3×1 vector expressing joint torques. q, \dot{q} and \ddot{q} are 3×1 vectors expressing joint position, velocity and acceleration respectively. Robot dynamics is derived manually and then verified using Matlab then it is applied and tested in Matlab and Simulink environment. Table 2 presents the whole mechanical properties of the robot at a glance. These properties include body segment length, body segment weight, moments of inertia with respect to center of mass and expressed in the coordinate system on the center of mass and center of mass expressed in the base coordinate system. For control purposes the state space model of the robot is expressed by the following equations:

$$\dot{X} = f(x) + g(x)\tau \quad (8)$$

$$f(x) = \begin{pmatrix} \dot{q} \\ -D^{-1}(q)(H(q, \dot{q}) + G(q)) \end{pmatrix} \quad (9)$$

$$g(x) = \begin{pmatrix} O_{n \times n} \\ D^{-1}(q) \end{pmatrix} \quad (10)$$

3. Control law formulation

Control algorithms applied on rehabilitation robots are designed considering two major aims: (1) passive rehabilitation in which the patient remains passive and the robot moves the patient's hand through a predefined trajectory and (2) active rehabilitation in which the patient initiates the movement and is partially assisted or resisted by the robotic device. In this paper the designed controller is intended to be used in passive rehabilitation. The proposed controller is Lyapunov-based

with integral action (Mahmood & Mhaskar, 2012). The designed controller should effectively track the desired trajectory and reject disturbance and other system uncertainties. The disturbance here is considered to be constant and bounded and all other system uncertainties are modeled as a constant bounded disturbance. Suppose \mathbf{d} is the disturbance torque. System dynamic model can be stated as follows:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{d}. \quad (11)$$

If $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ one can write the control input to the system as follows:

$$\boldsymbol{\tau} = \mathbf{D}(\mathbf{q})\ddot{\boldsymbol{\xi}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\boldsymbol{\xi}} + \mathbf{G}(\mathbf{q}) - \mathbf{K}_D\boldsymbol{\sigma} - \hat{\mathbf{d}}. \quad (12)$$

where, $\hat{\mathbf{d}}$ is an estimate of \mathbf{d} . If $\hat{\mathbf{d}} = \mathbf{d}$, then the disturbance input to the system will be rejected.

Table 2. Mechanical properties of the robot at a glance

Body segment	Segment (length m)	Segment weight (kg)	Moments of inertial ($kg.m^2$) with respect to center of mass and expressed in the center of mass coordinate frame			Center of mass (m) expressed in the base coordinate frame		
			I_{xx}	I_{yy}	I_{zz}	x	y	z
Shoulder joint(from point 1 in Fig.2. to shoulder joint placement)	0.21	3.8	0.0934	0.0511	0.0559	-0.193	0.0096	-0.174
Arm (from shoulder joint placement to point 2 in Fig.2.)	0.12	3.6	0.0728	0.0148	0.0164	-0.111	0.00471	-0.337
Limb and limb holder	0.035	4.5	0.0175	0.0621	0.0719	-0.0067	-0.044	-0.447

For now, it is supposed that $\hat{\mathbf{d}} \neq \mathbf{d}$. Estimation error can be defined as follows:

$$\tilde{\mathbf{d}} = \mathbf{d} - \hat{\mathbf{d}} \quad (13)$$

Substituting Eq. (12) in Eq. (11) and defining:

$$\boldsymbol{\sigma} = \dot{\mathbf{q}} - \dot{\boldsymbol{\xi}} \quad (14)$$

One can write:

$$\mathbf{D}(\mathbf{q})\dot{\boldsymbol{\sigma}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\sigma} + \mathbf{K}_D\boldsymbol{\sigma} = \tilde{\mathbf{d}}. \quad (15)$$

According to passivity theorem (Khalil, 2002) if mapping $-\boldsymbol{\sigma} \mapsto \tilde{\mathbf{d}}$ is passive with respect to some functions of V_1 and also $\tilde{\mathbf{d}}$ is bounded so $\mathbf{e} = \mathbf{q} - \mathbf{q}_d$ will be continuous thus \mathbf{e} and $\dot{\mathbf{e}}$ will asymptotically tend to zero. It means:

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = \lim_{t \rightarrow \infty} \dot{\mathbf{e}}(t) = 0 \quad (16)$$

Now it is shown that $\tilde{\mathbf{d}}$ is bounded and a proper estimation law is proposed for $\hat{\mathbf{d}}$. V is defined as follows:

$$V = \frac{1}{2}\boldsymbol{\sigma}^T \mathbf{D}(\mathbf{q})\boldsymbol{\sigma} + \frac{1}{2}\tilde{\mathbf{d}}^T \mathbf{K}_I^{-1}\tilde{\mathbf{d}}, \quad (17)$$

where, $\mathbf{K}_I = k_i \mathbf{I}_{3 \times 3}$ is a positive definite matrix. Differentiating V with respect to time yields:

$$\dot{V} = \boldsymbol{\sigma}^T \mathbf{D}(\mathbf{q}) \dot{\boldsymbol{\sigma}} + \frac{1}{2} \boldsymbol{\sigma}^T \dot{\mathbf{D}}(\mathbf{q}) \boldsymbol{\sigma} + \tilde{\mathbf{d}}^T \mathbf{K}_1^{-1} \dot{\tilde{\mathbf{d}}}. \quad (18)$$

Substituting Eq. (15) in Eq. (18) one can obtain:

$$\dot{V} = \boldsymbol{\sigma}^T (-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\sigma} - \mathbf{K}_b \boldsymbol{\sigma} + \tilde{\mathbf{d}}) + \frac{1}{2} \boldsymbol{\sigma}^T \dot{\mathbf{D}}(\mathbf{q}) \boldsymbol{\sigma} + \tilde{\mathbf{d}}^T \mathbf{K}_1^{-1} \dot{\tilde{\mathbf{d}}} = \frac{1}{2} \boldsymbol{\sigma}^T (\dot{\mathbf{D}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \boldsymbol{\sigma} - \boldsymbol{\sigma}^T \mathbf{K}_b \boldsymbol{\sigma} + \tilde{\mathbf{d}}^T (\boldsymbol{\sigma} + \mathbf{K}_1^{-1} \dot{\tilde{\mathbf{d}}}) = -\boldsymbol{\sigma}^T \mathbf{K}_b \boldsymbol{\sigma} + \tilde{\mathbf{d}}^T (\boldsymbol{\sigma} + \mathbf{K}_1^{-1} \dot{\tilde{\mathbf{d}}}). \quad (19)$$

Now if:

$$\boldsymbol{\sigma} + \mathbf{K}_1^{-1} \dot{\tilde{\mathbf{d}}} = \mathbf{0} \rightarrow \dot{\tilde{\mathbf{d}}} = \mathbf{K}_1 \boldsymbol{\sigma}. \quad (20)$$

In deriving (20), it is assumed that $\tilde{\mathbf{d}}$ is constant bounded so:

$$\dot{V} = -\boldsymbol{\sigma}^T \mathbf{K}_b \boldsymbol{\sigma} \leq 0. \quad (21)$$

Since V is lower-bounded ($V \geq 0$) and decreasing ($\dot{V} \leq 0$) so $\lim_{t \rightarrow \infty} V(t)$ is also bounded. Since $\frac{1}{2} \boldsymbol{\sigma}^T \mathbf{D}(\mathbf{q}) \boldsymbol{\sigma}$ and $\frac{1}{2} \tilde{\mathbf{d}}^T \mathbf{K}_1^{-1} \dot{\tilde{\mathbf{d}}}$ are non-negative matrices, $\mathbf{D}(\mathbf{q})$ and \mathbf{K}_1^{-1} are limited matrices so $\boldsymbol{\sigma}$ and $\tilde{\mathbf{d}}$ are bounded and this means that $\boldsymbol{\sigma}, \tilde{\mathbf{d}} \in L_\infty$. Using:

$$V(t) - V(0) \leq -\lambda_{\min}(\mathbf{K}_b) \int_0^t \|\boldsymbol{\sigma}(s)\|^2 ds. \quad (22)$$

One can conclude that $\boldsymbol{\sigma} \in L_2$; so \mathbf{e} and $\dot{\mathbf{e}}$ are bounded and that $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is bounded and one can conclude that $\dot{\boldsymbol{\sigma}} \in L_\infty$. Using Eq. (21):

$$\ddot{V} = -2\boldsymbol{\sigma}^T \mathbf{K}_b \dot{\boldsymbol{\sigma}}. \quad (23)$$

Because $\dot{\boldsymbol{\sigma}}, \boldsymbol{\sigma} \in L_\infty$, it can be concluded that $\ddot{V} \in L_\infty$. Using Barbalat's lemma one can show:

$$\begin{cases} \ddot{V} \in L_\infty \\ \lim_{t \rightarrow \infty} V(t) < \infty \end{cases} \rightarrow \lim_{t \rightarrow \infty} \dot{V}(t) = 0 \rightarrow \lim_{t \rightarrow \infty} \boldsymbol{\sigma} = 0 \quad (24)$$

Using Eq. (20):

$$-\boldsymbol{\sigma}^T \tilde{\mathbf{d}} = -\dot{\tilde{\mathbf{d}}}^T \mathbf{K}_1^{-1} \tilde{\mathbf{d}} = \dot{\tilde{\mathbf{d}}}^T \mathbf{K}_1^{-1} \tilde{\mathbf{d}}. \quad (25)$$

So:

$$\int_0^t -\boldsymbol{\sigma}^T(s) \tilde{\mathbf{d}}(s) ds = \int_0^t \dot{\tilde{\mathbf{d}}}^T(s) \mathbf{K}_1^{-1} \tilde{\mathbf{d}}(s) ds = \frac{1}{2} \tilde{\mathbf{d}}^T \mathbf{K}_1^{-1} \tilde{\mathbf{d}} \Big|_0^t = V_1(t) - V_1(0), \quad (26)$$

where, $V_1 = \frac{1}{2} \tilde{\mathbf{d}}^T \mathbf{K}_1^{-1} \tilde{\mathbf{d}}$; so mapping $-\boldsymbol{\sigma} \mapsto \tilde{\mathbf{d}}$ is passive with respect to some functions V_1 . It is proved that $\lim_{t \rightarrow \infty} \mathbf{e}(t) = \lim_{t \rightarrow \infty} \dot{\mathbf{e}}(t) = 0$. Then the control law can be stated as follows:

$$\tau = D(q)\ddot{\xi} + C(q, \dot{q})\dot{\xi} + G(q) - K_D\sigma - K_I \int_0^t \sigma(s)ds. \tag{27}$$

In summary the Lyapunov based controller with integral action can be given as follows:

$$\begin{aligned} \tau &= D(q)\ddot{\xi} + C(q, \dot{q})\dot{\xi} + G(q) - K_D\sigma - \hat{d} \\ \hat{d} &= K_I\sigma, \hat{d}(0) = 0 \\ \dot{\xi} &= \dot{q}_d - \Lambda e \\ \sigma &= \dot{q} - \dot{\xi} = \dot{e} + \Lambda e \\ e &= q - q_d \end{aligned} \tag{28}$$

Where,

$K_D = k_d I_{3 \times 3}$, $K_I = k_i I_{3 \times 3}$ and $\Lambda = \lambda I_{3 \times 3}$ are three positive definite matrices.

4. Simulation and results

For simulation, the desired trajectory is stated as $q_a = [0.5 \sin(t) \quad 0.5 \cos(t) + 1 \quad 0.5 \cos(t)]^T$ which is free of singular point. It is assumed that $k_d = 40$, $k_i = 20$ and $\lambda = 1$. The values are derived by trial and error to gain an optimum result. Simulation time is 20(s). Initial position vector is $q_0 = [0 \quad \frac{\pi}{3} \quad 0]^T$ that is in the range of motion (ROM) of the shoulder joint. Fig. 6 shows tracking performance of this controller for each of the joints. It can be seen that the performance of this controller in tracking desired trajectory is excellent. Fig. 7 demonstrates control inputs and Figure 8 shows tracking error of this controller. It can be seen from Fig. 8 that the error is considerably small and negligible and the controller is efficient in tracking desired trajectory. The control law has a term for gravity compensation that is very important in rehabilitation since there should be no load on human limb due to gravity. The proposed controller is robust, this means that changing the amplitude of uncertainty or robot parameters do not have affect in the performance of it. Rehabilitation exercises are carried out with different speeds. The controller also shows excellent performance in exercises with different speeds.

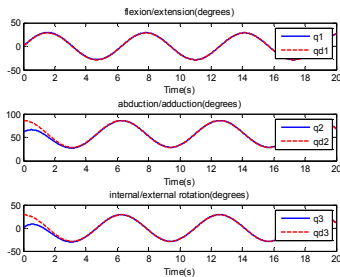


Fig. 6. Simulation results with Lyapunov-based controller with integral action in tracking desired trajectory for each joint

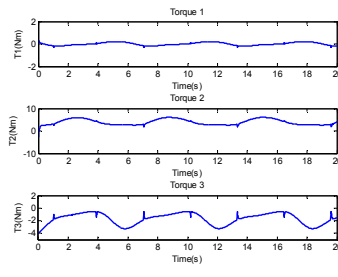


Fig. 7. Control inputs based on Lyapunov-based controller with integral action for each joint

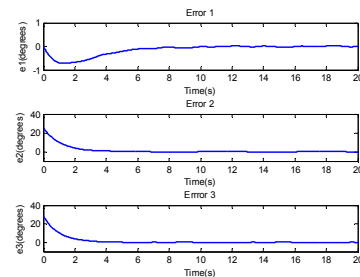


Fig. 8. Tracking error based on Lyapunov-based controller with integral action for each joint

5. Conclusion

In this paper mechanical design, simulation and control of a new exoskeleton robot for use in upper-limb rehabilitation after stroke was presented. The mechanical design was done and tested in Solidworks and kinematics and dynamics of the robot were derived using DH convention and generalized d'Alembert method respectively. A new circular open mechanism was proposed for joint 3 that is intended to solve the known issues with rehabilitation robots such as long wiring and discomfort associated with closed mechanisms. A control algorithm based on Lyapunov-based controller with integral action was proposed and applied on the robot. Simulation results with this controller showed effectiveness of this controller in tracking desired trajectory and rejecting constant bounded disturbance.

6. Future works

After obtaining successful results in the CAD environment and effectiveness of the controller in tracking desired trajectories, the robot will be made and experimental results will be published in the future. Active rehabilitation using impedance and admittance control will also be implemented.

References

- Brainin, M., Bornstein, N., Boysen, G., & Demarin, V. (2000). Acute neurological stroke care in Europe: results of the European Stroke Care Inventory. *European Journal of Neurology*, 7 (1), 5-10.
- Carignan, C., Liszka, M., & Roderick, S. (2005). Design of an arm exoskeleton with scapula motion for shoulder rehabilitation. In: *Advanced Robotics, ICAR'05. Proceedings., 12th International Conference on. IEEE*, pp 524-531.
- Fazekas, G., Horvath, M., Troznai, T., & Toth, A. (2007). Robot-mediated upper limb physiotherapy for patients with spastic hemiparesis: a preliminary study. *Journal of Rehabilitation Medicine*, 39 (7), 580-582.
- Frisoni, A., Borelli, L., Montagner, A., Marcheschi, S., Procopio, C., Salsedo, F., Bergamasco, M., Carboncini, M., Tolaini, M., & Rossi, B. (2007) Arm rehabilitation with a robotic exoskeleton in Virtual Reality. In: *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on, IEEE*, 631-642.
- Hesse, S., Schulte-Tiggas, G., Konrad, M., Bardeleben, A., & Werner, C. (2003). Robot-assisted arm trainer for the passive and active practice of bilateral forearm and wrist movements in hemiparetic subjects. *Archives of physical medicine and rehabilitation*, 84 (6), 915-920.
- Khalil, H.K. (2002). *Nonlinear systems*. Vol. 3: Prentice hall Upper Saddle River.
- Kwakkel, G., Kollen, B.J., & Krebs, H.I. (2008). Effects of robot-assisted therapy on upper limb recovery after stroke: a systematic review. *Neurorehabilitation and neural repair*, 22 (2), 111-121.
- Krebs, H.I., Hogan, N., Aisen, M.L., & Volpe, B.T. (1998). Robot-aided neurorehabilitation. *Rehabilitation Engineering, IEEE Transactions on*, 6 (1), 75-87.
- Lee, C., Lee, B., & Nigam, R. (1983). Development of the generalized d'Alembert equations of motion for mechanical manipulators. in *Decision and Control, 1983. The 22nd IEEE Conference on*, 1205-1210.
- Lloyd-Jones, D., Adams, R.J., Brown, T.M., Carnethon, M., Dai, S., De Simone, G., Ferguson, T.B., Ford, E., Furie, K., & Gillespie, C. (2010). Heart disease and stroke statistics—2010 update A report from the American Heart Association. *Circulation*, 121(7), e46-e215.
- Lum, P.S., Burgar, C.G., Van der Loos, M., Shor, P.C., Majmundar, M., Yap, R. (2006). MIME robotic device for upper-limb neurorehabilitation in subacute stroke subjects: A follow-up study. *Journal of rehabilitation research and development*, 43(5), 631.
- Loureiro, R., Amirabdollahian, F., Topping, M., Driessen, B., & Harwin, W. (2003). Upper limb robot mediated stroke therapy—GENTLE/s approach. *Autonomous Robots*, 15 (1), 35-51.
- Mahmood, M., & Mhaskar, P. (2012). Lyapunov-based model predictive control of stochastic nonlinear systems. *Automatica*, 48, 2271-2276.

- Mihelj, M., Nef, T., & Riener, R. (2007). ARMin II-7 DoF rehabilitation robot: mechanics and kinematics. In: *Robotics and Automation, IEEE International Conference, 2007. IEEE*, pp 4120-4125.
- Nakayama, H., Jørgensen, H., Raaschou, H., & Olsen, T.S. (1994). Recovery of upper extremity function in stroke patients: the Copenhagen Stroke Study. *Archives of physical medicine and rehabilitation*, 75 (4), 394.
- Nef, T., Mihelj, M., & Riener, R. (2007). ARMin: a robot for patient-cooperative arm therapy. *Medical & biological engineering & computing*, 45 (9), 887-900.
- Nef, T., Guidali, M., & Riener, R. (2009). ARMin III—arm therapy exoskeleton with an ergonomic shoulder actuation. *Applied Bionics and Biomechanics*, 6 (2), 127-142.
- Pons, J.L. (2008). Wearable robots: biomechatronic exoskeletons.
- Prange, G.B., Jannink, M.J., Groothuis-Oudshoorn, C.G., Hermens, H.J., & IJzerman, M.J. (2006). Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke. *Journal of rehabilitation research and development*, 43 (2), 171.
- Reinkensmeyer, D.J., Kahn, L.E., Averbuch, M., McKenna-Cole, A., Schmit, B.D., & Rymer, W.Z. (2000). Understanding and treating arm movement impairment after chronic brain injury: progress with the ARM guide. *Journal of rehabilitation research and development*, 37 (6), 653-662.
- Stienen, A.H., Hekman, E.E., Van der Helm, F.C., Prange, G.B., Jannink, M.J., Aalsma, A.M., & Van der Kooij, H. (2007). Dampace: dynamic force-coordination trainer for the upper extremities. In: *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on, 2007. IEEE*, pp 820-826.
- Sanchez, R.J., Liu, J., Rao, S., Shah, P., Smith, R., Rahman, T., Cramer, S.C., Bobrow, J.E., & Reinkensmeyer, D.J. (2006). Automating arm movement training following severe stroke: functional exercises with quantitative feedback in a gravity-reduced environment. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14 (3), 378-389.
- Siciliano, B., & Sciavicco, L. (2009). *Robotics: modelling, planning and control*: Springer Verlag.
- Thorvaldsen, P., Asplund, K., Kuulasmaa, K., Rajakangas, A-M., & Schroll, M. (1995). Stroke incidence, case fatality, and mortality in the WHO MONICA project. *Stroke*, 26 (3), 361-367.
- Toth, A., Fazekas, G., Arz, G., Jurak, M., & Horvath, M. (2005). Passive robotic movement therapy of the spastic hemiparetic arm with REHAROB: report of the first clinical test and the follow-up system improvement. In: *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on, IEEE*, 127-130.
- Winter, D.A. (2009). *Biomechanics and motor control of human movement*: John Wiley & Sons.

Appendix

In Eq. (7):

$$\mathbf{D}(\mathbf{q}) = \begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{pmatrix}, \quad \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{pmatrix} H_1 \\ H_2 \\ H_3 \end{pmatrix}, \quad \mathbf{G}(\mathbf{q}) = \begin{pmatrix} G_1 \\ G_2 \\ G_3 \end{pmatrix}$$

Let us define $m_i, i = 1, 2, 3$ link masses and inertia tensor matrix as:

$$\mathbf{I}_i = \begin{pmatrix} I_{ixx} & 0 & 0 \\ 0 & I_{iyy} & 0 \\ 0 & 0 & I_{izz} \end{pmatrix}, i = 1, 2, 3$$

Using Table 1 transformation matrices will be derived as follows:

$$\mathbf{T}_1^0 = \begin{pmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}_2^1 = \begin{pmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}_3^2 = \begin{pmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Where, c_i and s_i are $\cos(\theta_i)$ and $\sin(\theta_i)$ respectively. Let \mathbf{P}_i be the first three elements of the last column of \mathbf{T}_i^0 and

$\mathbf{r}_i = (r_{ix} \quad r_{iy} \quad r_{iz})^T$, $i = 1, 2, 3$ is link center of masses vector. Defining $\bar{\mathbf{c}}_i = \mathbf{r}_i - \mathbf{P}_{i-1}$, $i = 1, 2, 3$ one can obtain:

$$\begin{aligned} D_{11} &= I_{1yy} + m_1(r_{1x}^2 + r_{1y}^2) + m_2(r_{2x}^2 + r_{2y}^2) + m_3(r_{3x}^2 + r_{3y}^2) + (I_{2zz} + I_{3zz})c_2^2 + (I_{2xx} + I_{3xx}c_3^2 + I_{3yy}c_3^2)s_2^2 \\ D_{12} &= (I_{3yy} - I_{3xx})s_2s_3c_3 + m_2(r_{2x}c_{2z}s_1 - r_{2y}c_{2z}c_1) + m_3(r_{3x}c_{3z}s_1 - r_{3y}c_{3z}c_1) \\ D_{13} &= I_{3zz}c_2 + m_3((r_{3x}c_{3x} + r_{3y}c_{3y})c_2 - (r_{3x}c_{3z}c_1 + r_{3y}c_{3z}s_1)s_2) \\ D_{22} &= I_{2yy} + I_{3xx} + (I_{3yy} - I_{3xx})c_3^2 + m_2(c_{2y}^2 + c_{2z}^2 + c_1^2(c_{2x}^2 - c_{2y}^2) + 2c_{2x}c_{2y}c_1s_1) + m_3(c_{3y}^2 + c_{3z}^2 + 2c_{3x}c_{3y}c_1s_1 + c_1^2(c_{3x}^2 - c_{3y}^2)) \\ D_{23} &= m_3((c_{3x}^2s_1c_1 - 2c_{3x}c_{3y}c_1^2 + c_{3x}c_{3y} - c_{3y}^2s_1c_1)s_2 + (c_{3x}c_{3x}s_1 - c_{3y}c_{3z}c_1)c_2) \\ D_{33} &= I_{3zz} + m_3(c_{3x}^2(1 - c_1^2s_2^2) - 2c_{3x}c_{3y}c_1s_1s_2^2 - 2c_{3x}c_{3z}c_1c_2s_2 + c_{3y}^2c_1^2s_2^2 + c_{3y}^2c_2^2 - 2c_{3y}c_{3z}s_1s_2c_2 + c_{3z}^2s_2^2) \\ H_1 &= \dot{\theta}_1^2(m_1(c_{1x}r_{1y} - c_{1y}r_{1x}) + m_2(r_{2y}c_{2x} - r_{2x}c_{2y}) + m_3(c_{3x}r_{3y} - c_{3y}r_{3x})) + \dot{\theta}_1\dot{\theta}_2(2(I_{2xx} - I_{2zz} + I_{3yy} - I_{3zz})c_2s_2 + 2m_2(r_{2x}c_{2z}c_1 + r_{2y}c_{2z}s_1) + 2m_3(r_{3x}c_{3z}c_1 + r_{3y}c_{3z}s_1)) + \dot{\theta}_1\dot{\theta}_3(2(I_{3yy} - I_{3xx})c_3s_3s_2^2 + m_3(c_{3x}r_{3y} - r_{3x}c_{3y} + r_{3y}c_{3x}c_2 - c_{3y}r_{3x}c_2 - c_{3z}r_{3y}c_1s_2 + c_{3z}r_{3x}s_1s_2)) + \dot{\theta}_2^2(m_2(r_{2y}c_{2x}c_1^2 - r_{2x}c_{2y}s_1^2 + c_1s_1(r_{2y}c_{2y} - r_{2x}c_{2x})) + m_3(r_{3y}c_{3x}c_1^2 - r_{3x}c_{3y}s_1^2 + c_1s_1(r_{3y}c_{3y} - r_{3x}c_{3x}))) + (I_{3yy} - I_{3xx})c_2c_3s_3) + \dot{\theta}_2\dot{\theta}_3((I_{3xx} - I_{3yy} - I_{3zz})s_2 + 2(I_{3yy} - I_{3xx})s_2c_3^2 + m_3((c_{3y}r_{3x} + c_{3x}r_{3y})c_1s_1s_2 - c_{3x}r_{3x}s_1^2s_2 - c_{3y}r_{3y}c_1^2s_2)) + \dot{\theta}_3^2(m_3c_2(r_{3y}c_{3x} - r_{3x}c_{3y})) \\ H_2 &= \dot{\theta}_1^2((I_{2zz} - I_{2xx} + I_{3yy} + I_{3zz} - I_{3xx}c_3^2 + I_{3yy}c_3^2)c_2s_2 - m_2c_{2z}(c_{2y}s_1 + c_{2x}c_1) - m_3c_{3z}(c_{3x}c_1 + c_{3y}s_1)) + \dot{\theta}_1\dot{\theta}_3((I_{3xx} - I_{3yy} + I_{3zz} - 2I_{3xx}c_3^2 + 2I_{3yy}c_3^2)s_2 + 2m_3(c_{3z}^2s_2 - c_{3x}c_{3z}c_1c_2 - c_{3y}c_{3z}c_2s_1)) + \dot{\theta}_3^2(m_3((c_{3z}^2 - c_{3y}^2 - c_{3x}^2c_1^2 + c_{3y}^2c_1^2)c_2s_2 + c_{3y}c_{3z}s_1 + c_{3x}c_{3z}c_1 - 2c_{3x}c_{3z}c_1c_2^2 - 2c_{3y}c_{3z}c_2^2s_1 - c_{3x}c_{3y}c_1c_2s_1s_2)) + \dot{\theta}_2\dot{\theta}_3(2(I_{3xx} - I_{3yy})c_3s_3 + 2m_3(c_{3x}c_{3y}c_2 + c_{3y}c_{3z}c_1s_2 - c_{3x}c_{3z}s_1s_2 - 2c_{3x}c_{3y}c_1^2c_2 + c_{3x}^2c_1c_2s_1 - c_{3y}^2c_1c_2s_1)) \\ H_3 &= \dot{\theta}_1^2((I_{3xx} - I_{3yy})c_3s_2^2s_3 + m_3(c_{3y}c_{3z}c_1s_2 - c_{3x}c_{3z}s_1s_2)) + \dot{\theta}_1\dot{\theta}_2((I_{3xx} - I_{3yy} - I_{3zz} - 2I_{3xx}c_3^2 + 2I_{3yy}c_3^2)s_2 + 2m_3c_{3z}(c_{3y}c_2s_1 + c_{3x}c_1c_2 - c_{3z}s_2)) + \dot{\theta}_2^2((I_{3yy} - I_{3xx})c_3s_3 + m_3((c_{3y}^2 - c_{3x}^2)c_1s_1c_2 + c_{3x}c_{3y}c_1^2c_2 + c_{3x}c_{3z}s_1s_2 - c_{3y}c_{3z}c_1s_2 - c_{3x}c_{3y}c_2)) \\ G_1 &= 0 \\ G_2 &= -g_0(m_2(c_{2x}c_1 + c_{2y}s_1) + m_3(c_{3x}c_1 + c_{3y}s_1)) \\ G_3 &= g_0m_3s_2(-c_{3x}s_1 + c_{3y}c_1) \end{aligned}$$