TAF Journal of Advances in Technology and Engineering Studies JATER

2017, **3**(6): 245-256

PRIMARY RESEARCH

Development and implementation of a sit-to-stand motion algorithm for humanoid robots

Fatma Gongor [1], **Onder Tutsoy** [2*], **Sule Colak** [3]

[1, 2, 3] Department of Electrical and Electronic Engineering, Adana Science and Technology University, Adana, Turkey

Index Terms
Center of Mass Estimation
Head-Arm-Torso System
Horizontal Distance
Initial Position
Joint Angle

Received: 16 October 2017
Accepted: 6 November 2017
Published: 4 December 2017

***Abstract*—** Humanoid robots are extensively becoming an essential part of the social life with the development and implementation of Human-Like behaviors such as walking, grasping, standing up, and sitting down. For generating natural Human-Like behaviors, the desired motion should be controlled properly in the dynamic and unstructured environments. This paper presents a Sit-to-Stand (STS) motion algorithm developed to lead autonomous Human-Like motion. The proposed algorithm has two main steps; where in the first one, horizontal distances of the humanoid robot for sitting down or standing up motions are identified to determine the corresponding joint angles, and in the second stage, the initial joint angles of the robot have been driven to the target joint angle states. Although this algorithm can be adopted to all humanoid robots, in this paper, it has been specifically applied to the NAO humanoid robot. Results show that the NAO humanoid robot is able to autonomously achieve halfway standing up and sitting down from chairs having heights between 9.5 and 11.5 cm. The developed STS algorithm will be combined with chair recognition and selection algorithms in the future, which will significantly enhance the capabilities of autonomous humanoid robots.

I. INTRODUCTION

Humans are able to achieve a variety of whole-body movements. These movements are affected by the dynamic environment and, therefore, they require appropriately co-ordinated relative arrangements of the body parts, especially legs and arms. For example, proper relative movements of the legs and arms are crucially important for sitting down and standing up for humans, and also for humanoid robots. Moreover, for successful sitting down and standing up behaviours, balance controlling on two feet and placement of the body's Center of Mass (CoM) in a stability region must be accomplished. This paper focuses on developing and analysing STS motion algorithm for humanoid robots, which will lead to more sophisticated autonomous humanoid robot design.

A number of STS analysis algorithms have been developed in the literature and they lead to important improvements in rehabilitation robotics [1], social robotics [2], and medical robotics. After 2010, research on STS motion algorithm development and analysis have gained impetus. Mistry [3], Gu [4], Pchelkin *et al*. [5], and Sugisaka [6] developed STS motion algorithms to control the humanoid robot having artificial muscles, links, and joints. It is stated that due to small support polygon areas of the humanoid robots, achieving sitting down motion for them becomes challenging [7, 8, 9]. Moreover, insufficient ankle torque produced caused problem during standing up and for addressing this problem, smooth trajectory planning and motion control are implemented together [5].

During the past decade, a great number of researches focused on developing various STS motion control algorithms where a number of them are implemented experimentally in real life such as helping elderly people [10, 11] and pathological patients [12, 13, 14]. Riley *et al*. [7] proposed a STS algorithm, which initially positions the CoM of the upper body in the support polygon area and then gen-

[*] Corresponding author: Onder Tutsoy
[†] Email: otutsoy@adanabtu.edu.tr

erates the required torques to drive the joints as desired. Recently, Coghlin & McFadyen [13], assessed STS motion of people having low back pain and compared their motions with normal people in control group. They stated that people having low back pain must generate larger knee torques to yield vertical trunk position during STS motion or they can increase the hip torques while reducing the knee torques. This research enlightens the STS motion algorithm design for humanoid robots.

During the STS motion, it is crucial to precisely follow the planned joint angle trajectories in the case of various unexpected changes. Bahar *et al*. [15], Prinz *et al*. [16] expressed that the key challenge of the STS algorithms is to control the whole body and its reactions to unplanned variations [16]. Therefore, the STS algorithms must track the planned trajectories together with action selection in case of variations [17]. While action selection concerns with generating the most appropriate control actions for different robot behaviors, the tracking ensures accuracy of the planned robot motion. Various action selection algorithms have been introduced in the literature [15] where one of them simply proposed IF-THEN rules based on the knee joint flexion to guarantee stability of the robot during STS motion [18]. This approach can be successful in certain cases, but since the real environment is highly dynamic, considering each unexpected case is not possible. Thus, more advanced control approaches should be developed to create utterly autonomous humanoid robots.

In the rest of the paper; Section 2 presents the CoM estimation technique used to provide joint angles for sitting down and standing up behaviours, Section 3 briefly reviews stability strategies to stabilize the STS motion, Section 4 illustrates NAO humanoid robot configuration, Section 5 analyses the simulation and experimental results and finally, conclusion and possible future works are stated in Section 6.

A. Proposed and Experimentally Implemented Sit-To-Stand Motion Analysis Algorithm

The proposed STS motion algorithm has two key stages as shown in Figure 1. In the first stage, Horizontal distances and their corresponding joint angles are determined. In the second stage, the joints of the robot are driven from its initial positions to final desired positions.

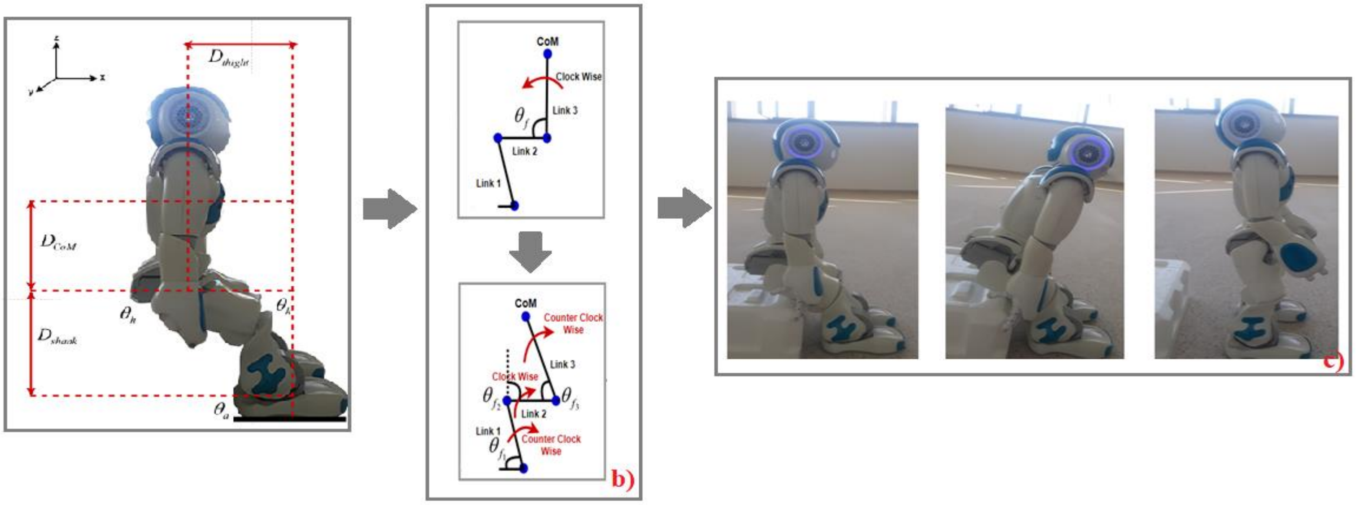


Fig. 1 . Proposed sit-to-stand motion analysis algorithm stages. a) Horizontal distance measurement identification, b) Joint angle determination, c) The STS motion algorithm application to NAO humanoid robot

II. ROBOT COM TRANSFERRING PHASES

In this section, the joint trajectories of the robot movement are planned based on Alexander STS technique, which focuses on decreasing the joint torques needed as the desired joint positions are approached, so that moderate control is achieved. Since planning a successful STS motion without falling requires accurate knowledge of the robot's CoM position, its calculation is performed when the robot is at sitting position. At this position, basically, the hips are supported by the chair which brings that thigh and shank masses are neglected to initiate the standing-up position. The purpose of CoM transferring phase is to bring the Head-Arm-Torso (HAT) CoM into the support polygon area that facilitates the stabilization strategy. Based on the position of the CoM, horizontal distance measurement and joint angles' determination are performed.

A. Horizontal Distance Measurement

In the first step of the CoM transeferring phase, the horizontal distance between the HAT CoM of the robot and the ankle joint is measured, and then change of the angles at hip and ankle joint is determined. The horizontal distance (*xi*)(in *x*-axis) is determined as;

$$x_i = \pm[\alpha_{hip}] + [\alpha_{knee}] \pm [\alpha_{ankle}] \tag{1}$$

Where:
$\alpha_{hip} = sin(|diff(\theta_{sh}, \theta_h)|)X^{D_{CoM}}$
$\alpha_{knee} = cos(|diff(\theta_{sk}, \theta_k)|)X^{D_{thigh}}$
$\alpha_{ankle} = sin(|diff(\theta_{sa}, \theta_a)|)X^{D_{shank}}$

The different term is used to identify the difference between hip, knee, and ankle joint positions obtained from sensors, $\theta_{sh}, \theta_{sk}, \theta_{sa}$ are hip, knee, and ankle joint positions, respectively, and $\theta_h, \theta_k, \theta_a, D_{CoM}$ is the distance between hip joint and the HAT CoM ,respectively D_{thigh}, D_{shank} represent the distances of thigh and shank, respectively, as shown in Figure 2.

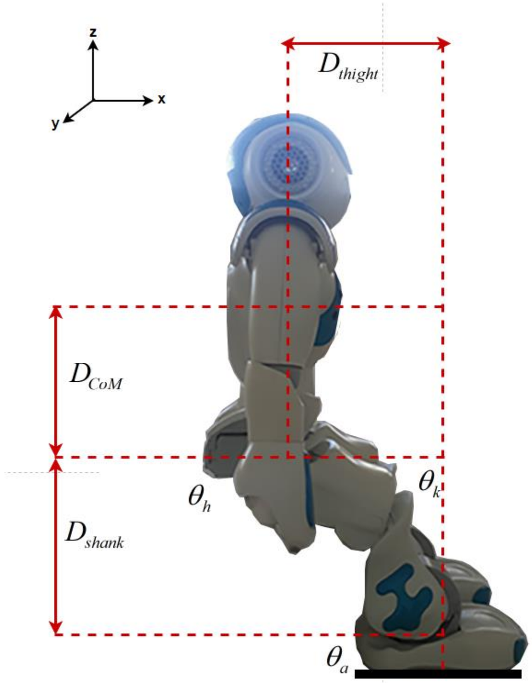


Fig. 2 . The configuration of NAO robot at sitting position

General joint angles and link distances for standard Nao humanoid robot at sitting position are as follows:

$$\begin{aligned} \theta_h &= -75.5\ \ degree \\ \theta_k &= 90\ \ degree \\ \theta_a &= -70\ \ degree \\ D_{CoM} &= 15\ \ cm \\ D_{thigh} &= 10\ \ cm \\ D_{shank} &= 10.3\ \ cm \end{aligned} \tag{2}$$

Basically with equation (1), the x_i position of HAT CoM can be obtained for any robot's default sitting position values. The HAT CoM position can be updated in terms of hip and knee joint angles as;

$$\theta_{hnew} = \theta_h + [\theta_k - \theta_{sk}] \tag{3}$$

ISSN: 2414-4592
DOI: 10.20474/jater-3.6.4

B. Joint Angle Determination

The CoM x_i position value is used to determine the change in joint angles for each joint.

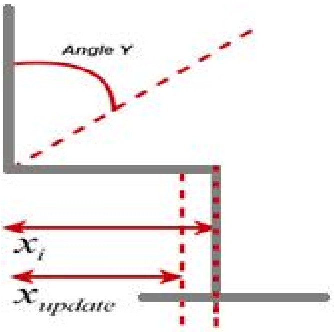


Fig. 3 . Configuration of the CoM position and hip angle at sitting position

The joint angles must be available to obtain the exact position of the HAT CoM and it must be ensured that it is in the support polygon area to guarantee the posture stability during STS motion implementation. During standing up motion, initially the robot body comes to the front and in this case, required hip joint angle is obtained as;

$$\theta_{h,update} = \theta_{sk} - \left[90 - abs \ \left(cos^{-1} \left(\frac{X_{update}}{D_{CoM}} \right) \right) \right] \quad (4)$$

If the updated hip joint angle $\theta h, update$ is larger than the limits of the hip joint angle, the new hip joint angle must be set to be equal to the maximum hip joint angle as;

$$\theta_{h,update} = \theta_{h,max}. \quad (5)$$

If the updated hip joint value $\theta_{h,update}$ is smaller, the horizontal distance measurement between the HAT CoM position and edge of support polygon will be updated as;

$$\theta_{h,new} = \theta_{h,update}.x_{new} \quad (6)$$

Notice that in this case, x_{update} is used to represent horizontal distance measurement value between the HAT CoM positions within the support polygon area.

After the determination of the hip joint angle, at the next step, the ankle joint is obtained and based on the limits of the hip joint, joint angle of the ankle must be updated. In this case, the residual distance measurement between the HAT CoM and support polygon area is calculated as;

$$x_{residual} = x_{new} - abs \ \left[D_{CoM} x^{cos} (90 - (\theta_{sh} - \theta_{h,new})) \right]$$
(7)

The residual distance measurement determines whether the ankle joint change is required. If the has a positive value, an updated ankle joint can be calculated as;

$$\theta_{\alpha,update} = \theta_{Sa} + \left[- \left(sin^{-1} \left(\frac{X_{residual}}{D_{shank}} \right) \right) - abs(\theta_{\alpha} - \theta_{Sa}) \right]$$
(8)

With these updated hip and ankle joint angles, the robot moves the desired standing position. The hip and ankle joint trajectories are generated using a cubic polynomial function. This function decreases the joint speed at the beginning and terminal parts of the joint trajectories to provide a soft transition. This cubic function given by Equation (9) is used to generate trajectories for the hip, knee, and ankle joints.

$$y(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 \quad (9)$$

The following results are obtained:

$\alpha_0 = \theta_{Sh,a,k}$
$\alpha_1 = 0$
$\alpha_2 = \frac{3}{t_f^2} (\theta_{Sh,a,k} - \theta_{h,k,a.update})$

$\alpha_3 = \frac{2}{t_f^3} (\theta_{Sh,a,k} - \theta_{h,k,a.update})$

$\theta_{Sh,a,k}$ term is used to determine angle values at the sitting position for the hip, knee, and ankle joints. $\theta_{h,k,a.update}$ represents the destination of each joint based on the updated angle which is obtained from the angle determination stage. When the hip and ankle joints are rotated to the new angle, the knee joint remains at the same position. In the first case, the motion is started with the hip joint and then the ankle joint also starts to rotate.

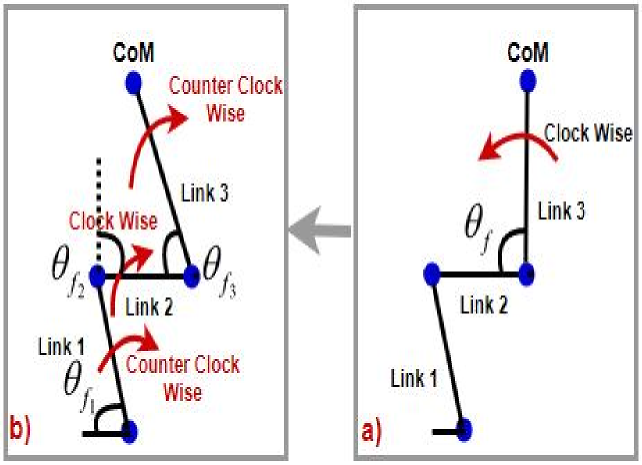


Fig. 4 . Illustration of STS cycle. a) CoM transferring phase, b) standing position phase

ISSN: 2414-4592
DOI: 10.20474/jater-3.6.4

In CoM transfer step, only the Link 3 moves in the clockwise direction until the CoM is located in the support polygon area. θ_f is used to describe the initial angle (90 degree) with respect to the Link 2. After completing the CoM transferring phase, it continues to the standing position step. During this phase, Link 3 and Link 1 move in counter-clockwise direction and at the same time, the Link 2 moves in clockwise direction. All movements will be terminated when the robot achieves the standing task. During the standing position; $\theta_{f_1}, \theta_{f_2}$ and θ_{f_3} terms are used to describe each joint's initial angle values with respect to the sole.

III. STABILIZATION STRATEGY

After determining each CoM's transfer phases, the second stage of the STS motion is to provide a stability region for the robot. In this case, controller input is Centre of Pressure (CoP), and this controller undergoes specific region boundary rules to find the exact direction and gain. The gain and rules are based on CoP position in three types of region as depicted in Figure 5.

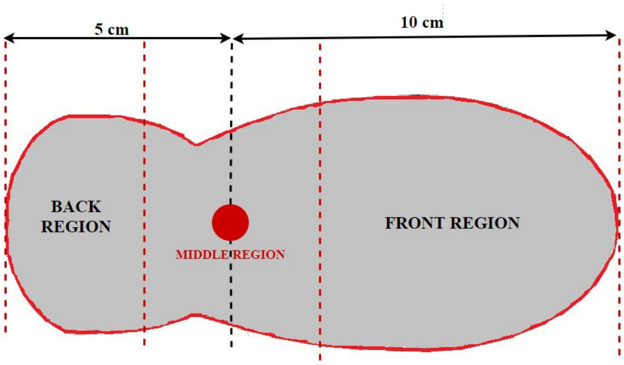


Fig. 5 . Regions defined at robot foot base on CoP position

The robot is defined as stable when the CoP is in the Middle Region and becoming unstable when the CoP is in the Back Region and Front Region as shown in Figure 5. Thus, the specific boundaries are set for each region as follows:

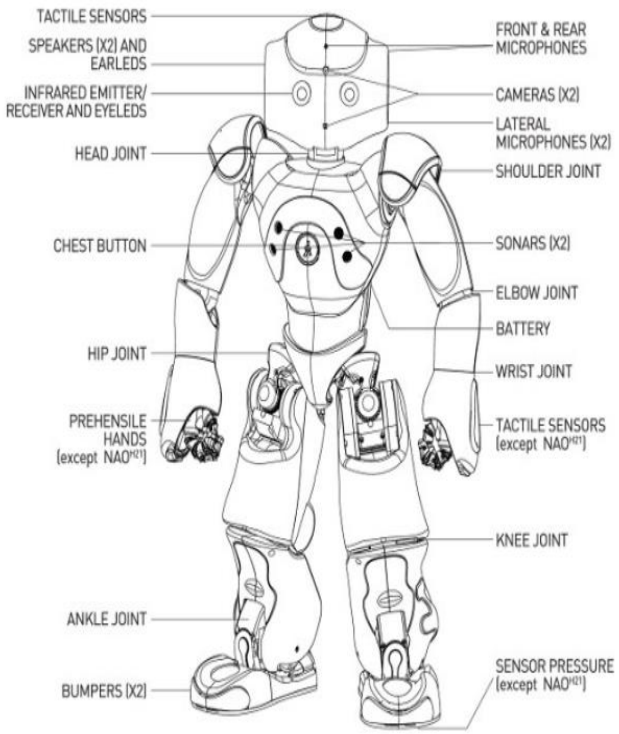
TABLE 1
REGION BOUNDARIES

Regions	Region Boundries (If - then Rules)
Back	CoP < -0.02m
Middle	(-0.02) < CoP <0.02m
Front	CoP > 0.02m

As can be seen from Table 1, 0.02m is used as a distance measurement value because the Middle Region represents a stable region and this region cannot be close to the ankle joint to avoid over sensitivity. The value 0.02m is also used as the front edge to represent the boundary between the Middle Region and the Front Region.

IV. NAO HUMANOID ROBOT CONFIGURATION

In this paper, NAO humanoid robot has been used for experimental purpose. Totally, there are 25 motors for controlling the movements of the Nao Humanoid Robot [19]. All movements are controlled by these motors. Three different types of motors are utilized to drive the NAO humanoid robot joints. First type of motors is used to actuate the legs, second type of motors is used to actuate the hands, and third one is used to actuate the arms and head. There are 14 different joints as shown in Figure 6(a). The other 11 joints are the arm and leg joints for the right side, controlling the shoulder, elbow, wrist, hand, hip, knee, and ankle [15]. HeadYaw, HeadPitch, and HipYawPitch are the only 3 motors that do not have both left and right side directions [15]. The joints can be controlled by either entering the desired trajectory values until the joint has reached the desired position.



ISSN: 2414-4592
DOI: 10.20474/jater-3.6.4

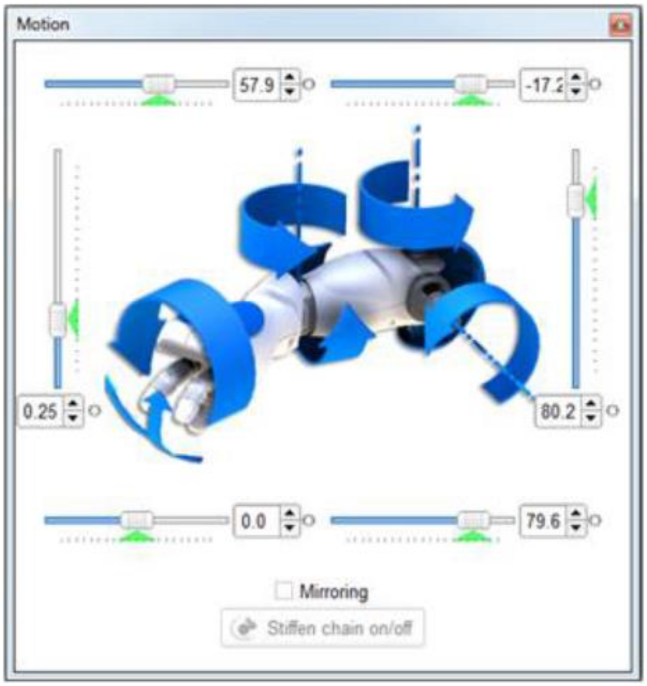


Fig. 6 . a) Labels of each joint [20], b) Motor values for each joint [21]

The Figure 6(b) shows right hand motion configuration and these boxes are used to edit the position of each motor.

In this paper, NAO capabilities are limited to collision detection, distance estimation, and physics engine. There are three different kinds of distance, which are used in the application throughout this work:

1) The distance between the robot head and the floor surface to determine whether the robot falls down.

2) The distance between the robot hips and the chair surface to determine whether the robot is on the chair. Notice that monitoring this distance is important to make sure hips are in contact with the chair.

3) The distance between the robot hips and the floor surface to measure whether the robot is sitting on the chair.

V. SIMULATION AND EXPERIMENTAL RESULTS

This section evaluates the STS motion analysis algorithm both in simulation and experimental environments. For analysing STS motion, a chair model is created as shown in Figure 7. The height of chair is varied from 9.25 cm to 11.35 cm.

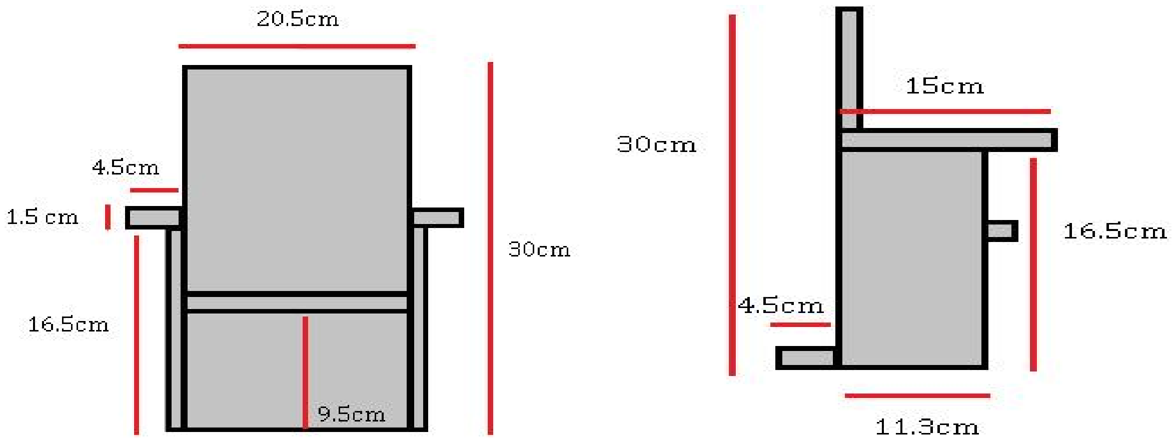


Fig. 7 . Front and side elevation of chair

The shank length (D_{shank}) is 100% equivalent to 10 cm where the shank is perpendicular to the ground. For ensuring the height is consistent, knee joint angle at initial position is varied to represent the chair height (90 degree is 10.0 cm). The STS motion is done until the end of CoM transferring phase where the robot is halfway from the standing position as shown in Figure 15b. This should be done to ensure that the obtained result is not influenced by the stabilization stage. With a new ankle joint $\theta_{\alpha,update}$ and $\theta_{h,update}$ calculation, the robot can achieve the target position without falling within chair height between 11.35 cm and 10 cm.

In this application, all steps are represented with boxes where each box represents a function and connection of each box leads to whole STS algorithm implementation to NAO humanoid robot. In this paper, it is only concentrated on STS motion, relying on coordinated movements of 3 joints, namely hip, knee, and ankle. For analysing motion of these three joints in Choregraphe, together with leg joints, arm joints having role in balancing during sitting down and standing up are also described as in Table 2. For analysing STS motion, each joint's motion ranges are described as in the following figures.

TABLE 2
JOINTS TO BE CONTROLLED WITH STS MOTION ALGORITHM

Blcok	Used Joints
Left arm	L shoulder roll, L shoulder pitch, L elbow yaw, L wrist yaw, L elbow roll
Right arm	R shoulder roll, R shoulder pitch, R elbow yaw, R wrist yaw, R elbow roll
Left leg	L hip yaw pitch, L hip pitch, L knee pitch, L ankle pitch, L ankle roll, L hip roll
Right leg	R hip yaw pitch, R hip pitch, R knee pitch, R ankle pitch, R ankle roll, R hip roll

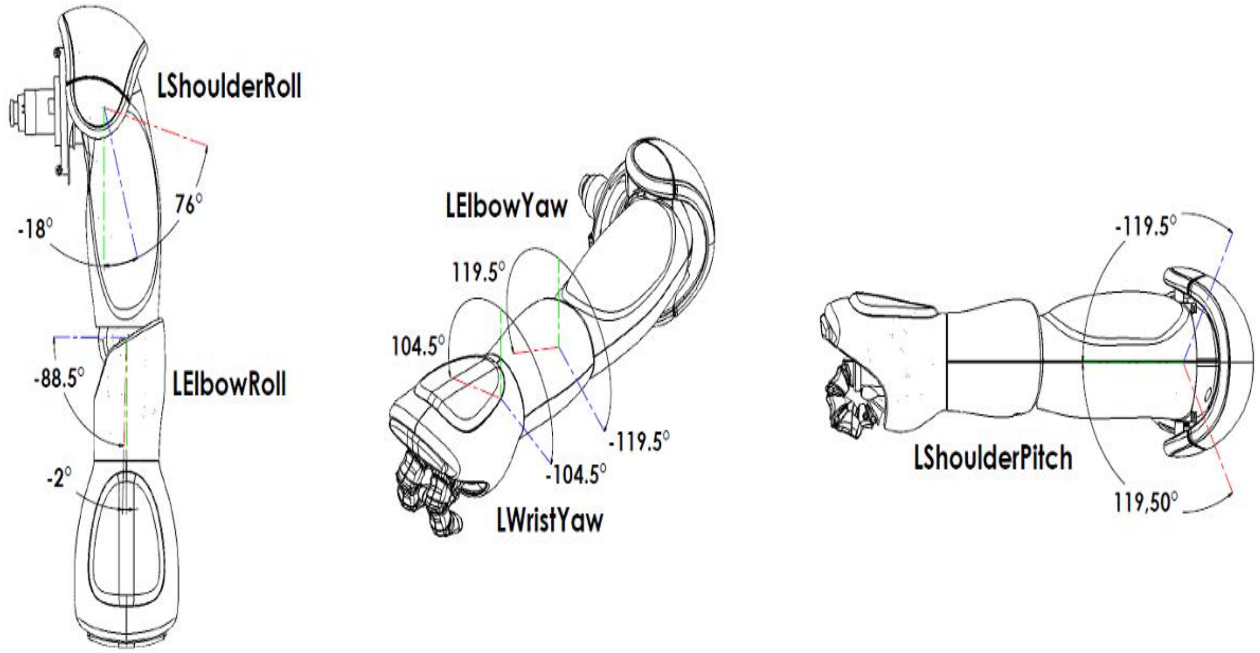


Fig. 8 . Lateral and sagittal view of left arm joint [21]

TABLE 3
THE RANGE OF THE MOTION FOR LEFT ARM JOINTS [21]

Joint name	Motion	Range (Radians)	Range (Degrees)
L shoulder pitch	Left shoulder joint front and back (Y)	(-119.5) to 119.5	(-2.0857) to 2.0857
L shoulder roll	Left shoulder joint righ and left (Z)	(-18) to 76	(-0.3142) to 1.3265
L elbow yaw	Left shoulder joint twist (X)	(-119.5) to 119.5	(-2.0857) to 2.0857
L elbow roll	Left elbow joint (Z)	(-88.5) to -2	(-1.5446 to 0.0349)
L wrist yaw	Left wrist joint (X)	(-104.5) to 104.5	(-1.8238) to 1.8238
L hand	Left hand	Open and close	Open and close

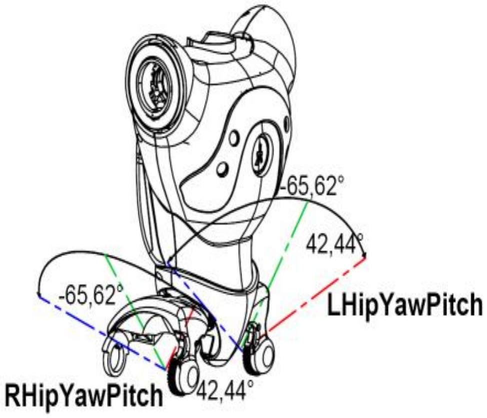


Fig. 9 . Lateral view of hip joint [21]

TABLE 4
THE RANGE OF THE MOTION FOR HIP JOINTS [19]

Joint name	Motion	Range (Radians)	Range (Degrees)
Left hip yaw pitch	Left hip joint twist (Y-Z 45^0)	-65.62 to 42.44	(-1.145) to 0.740
Right hip yaw pitch	Right hip joint twist (Y-Z 45^0)	-65.62 to 42.44	(-1.145) to 0.740

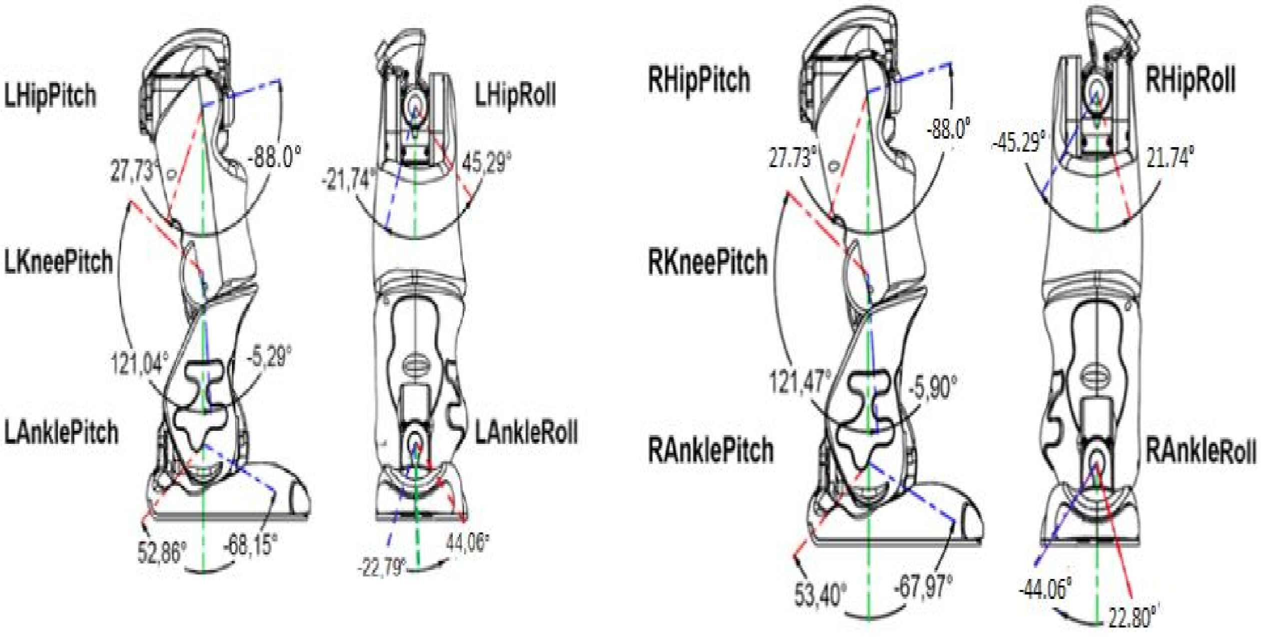


Fig. 10 . Lateral and sagittal view of left and right legs [21]

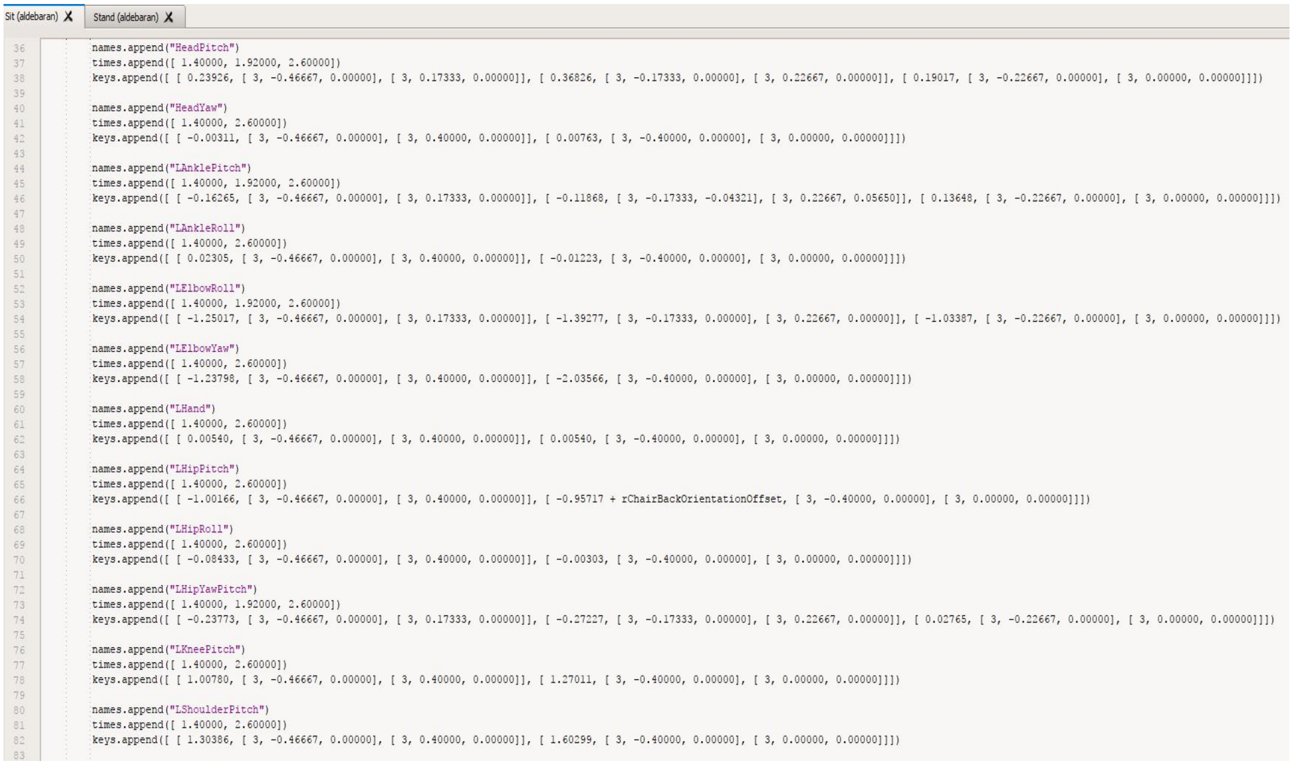
TABLE 5
THE RANGE OF THE MOTION FOR LEFT LEG JOINTS [21]

Joint name	Motion	Range (Radians)	Range (Degrees)
L hip roll	Left hip joint right and left (*X*)	(-21.74 to 45.29)	(-0.379 to 0.790)
L hip pitch	Left hip joint front and back (*Y*)	(-88.00 to 27.73)	(-1.535 to 0.4840)
L knee pitch	Left knee joint (*Y*)	(-5.29 to 121.04)	(-0.092 to 2.112)
L ankle pitch	Left ankle joint front and back (*Y*)	(-68.15 to 52.86)	(-1.189 to 0.922)
L ankle roll	Left ankle joint right and left (*X*)	(-22.79 to 44.06)	(-0.397 to 0.769)

TABLE 6
THE RANGE OF THE MOTION FOR RIGHT LEG JOINTS [21]

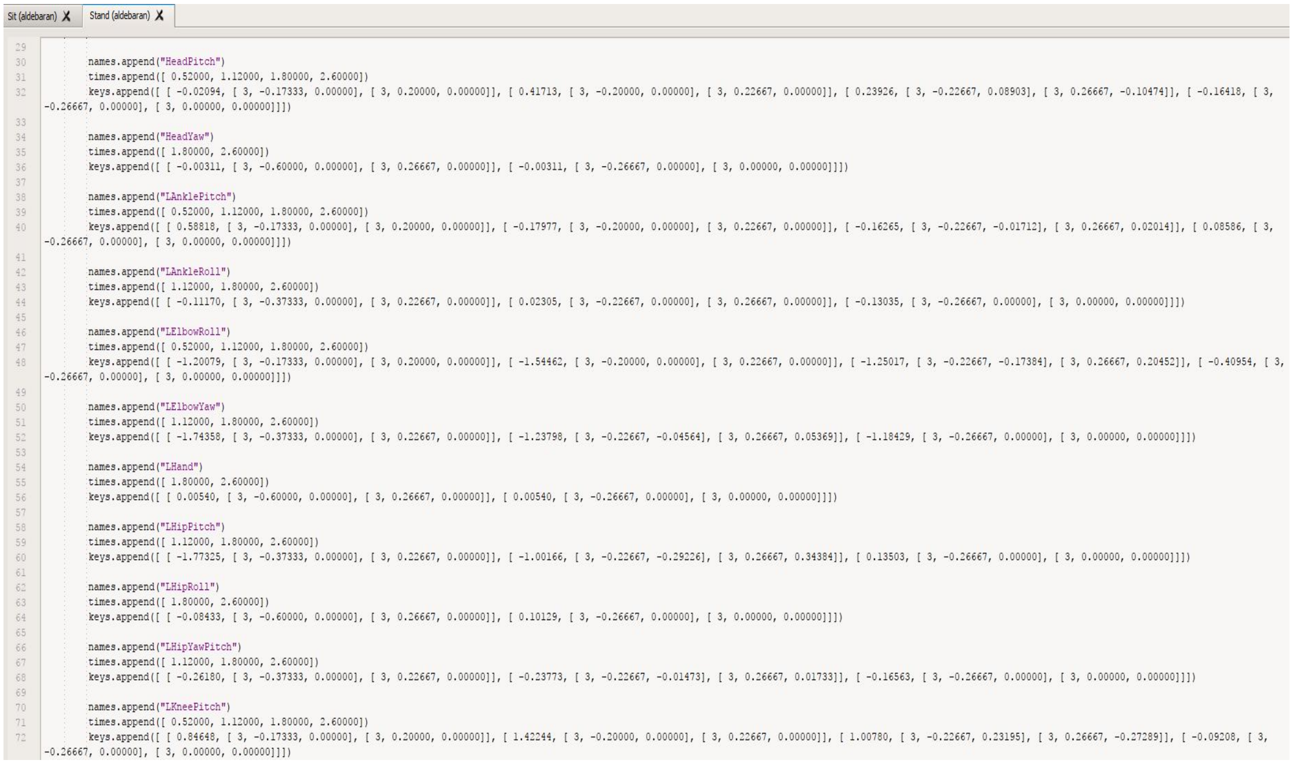
Joint name	Motion	Range (Radians)	Range (Degrees)
R hip roll	Right hip joint right and left (*X*)	(-45.29 to 21.74)	(-0.790 to 0.379)
R hip pitch	Right hip joint front and back (*Y*)	(-88.00 to 27.73)	(-1.535 to 0.484)
R knee pitch	Right knee joint (*Y*)	(-5.90 to 121.47)	(-0.103 to 2.120)
R ankle pitch	Right ankle joint front and back (*Y*)	(-67.97 to 53.40)	(-1.186 to 0.932)
R ankle roll	Right ankle joint right and left (*X*)	(-44.06 to 22.80)	(-0.768 to 0.397)

Furthermore, each block's joint values are described in Choregraphe environment as depicted in Figures 11 and 12. To experiment STS motion algorithm on NAO humanoid robot through Choregraphe; initially, implementation of the algorithm is activated by touching the robot's head. Then the robot can be manipulated easily to obtain desired sitting down and standing up positions as shown in the Figure 13. As can be seen in Figure 14, desired trajectories obtained with STS motion algorithm are applied to NAO humanoid robot in real life experiment.



```
names.append("HeadPitch")
times.append([ 1.40000, 1.92000, 2.60000])
keys.append([ [ 0.23926, [ 3, -0.46667, 0.00000], [ 3, 0.17333, 0.00000]], [ 0.36826, [ 3, -0.17333, 0.00000], [ 3, 0.22667, 0.00000]], [ 0.19017, [ 3, -0.22667, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("HeadYaw")
times.append([ 1.40000, 2.60000])
keys.append([ [ -0.00311, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ 0.00763, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LAnklePitch")
times.append([ 1.40000, 1.92000, 2.60000])
keys.append([ [ -0.16265, [ 3, -0.46667, 0.00000], [ 3, 0.17333, 0.00000]], [ -0.11868, [ 3, -0.17333, -0.04321], [ 3, 0.22667, 0.05650]], [ 0.13648, [ 3, -0.22667, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LAnkleRoll")
times.append([ 1.40000, 2.60000])
keys.append([ [ 0.02305, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ -0.01223, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LElbowRoll")
times.append([ 1.40000, 1.92000, 2.60000])
keys.append([ [ -1.25017, [ 3, -0.46667, 0.00000], [ 3, 0.17333, 0.00000]], [ -1.39277, [ 3, -0.17333, 0.00000], [ 3, 0.22667, 0.00000]], [ -1.03387, [ 3, -0.22667, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LElbowYaw")
times.append([ 1.40000, 2.60000])
keys.append([ [ -1.23798, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ -2.03566, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LHand")
times.append([ 1.40000, 2.60000])
keys.append([ [ 0.00540, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ 0.00540, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LHipPitch")
times.append([ 1.40000, 2.60000])
keys.append([ [ -1.00166, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ -0.95717 + rChairBackOrientationOffset, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LHipRoll")
times.append([ 1.40000, 2.60000])
keys.append([ [ -0.08433, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ -0.00303, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LHipYawPitch")
times.append([ 1.40000, 1.92000, 2.60000])
keys.append([ [ -0.23773, [ 3, -0.46667, 0.00000], [ 3, 0.17333, 0.00000]], [ -0.27227, [ 3, -0.17333, 0.00000], [ 3, 0.22667, 0.00000]], [ 0.02765, [ 3, -0.22667, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LKneePitch")
times.append([ 1.40000, 2.60000])
keys.append([ [ 1.00780, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ 1.27011, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])

names.append("LShoulderPitch")
times.append([ 1.40000, 2.60000])
keys.append([ [ 1.30386, [ 3, -0.46667, 0.00000], [ 3, 0.40000, 0.00000]], [ 1.60299, [ 3, -0.40000, 0.00000], [ 3, 0.00000, 0.00000]]])
```

Fig. 11 . Used joints' representation in choregraphe (for sitting position)



```
        names.append("HeadPitch")
        times.append([ 0.52000, 1.12000, 1.80000, 2.60000])
        keys.append([ [ -0.02094, [ 3, -0.17333, 0.00000], [ 3, 0.20000, 0.00000]], [ 0.41713, [ 3, -0.20000, 0.00000], [ 3, 0.22667, 0.00000]], [ 0.23926, [ 3, -0.22667, 0.08903], [ 3, 0.26667, -0.10474]], [ -0.16418, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("HeadYaw")
        times.append([ 1.80000, 2.60000])
        keys.append([ [ -0.00311, [ 3, -0.60000, 0.00000], [ 3, 0.26667, 0.00000]], [ -0.00311, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LAnklePitch")
        times.append([ 0.52000, 1.12000, 1.80000, 2.60000])
        keys.append([ [ 0.58818, [ 3, -0.17333, 0.00000], [ 3, 0.20000, 0.00000]], [ -0.17977, [ 3, -0.20000, 0.00000], [ 3, 0.22667, 0.00000]], [ -0.16265, [ 3, -0.22667, -0.01712], [ 3, 0.26667, 0.02014]], [ 0.08586, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LAnkleRoll")
        times.append([ 1.12000, 1.80000, 2.60000])
        keys.append([ [ -0.11170, [ 3, -0.37333, 0.00000], [ 3, 0.22667, 0.00000]], [ 0.02305, [ 3, -0.22667, 0.00000], [ 3, 0.26667, 0.00000]], [ -0.13035, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LElbowRoll")
        times.append([ 0.52000, 1.12000, 1.80000, 2.60000])
        keys.append([ [ -1.20079, [ 3, -0.17333, 0.00000], [ 3, 0.20000, 0.00000]], [ -1.54462, [ 3, -0.20000, 0.00000], [ 3, 0.22667, 0.00000]], [ -1.25017, [ 3, -0.22667, -0.17384], [ 3, 0.26667, 0.20452]], [ -0.40954, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LElbowYaw")
        times.append([ 1.12000, 1.80000, 2.60000])
        keys.append([ [ -1.74358, [ 3, -0.37333, 0.00000], [ 3, 0.22667, 0.00000]], [ -1.23798, [ 3, -0.22667, -0.04564], [ 3, 0.26667, 0.05369]], [ -1.18429, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LHand")
        times.append([ 1.80000, 2.60000])
        keys.append([ [ 0.00540, [ 3, -0.60000, 0.00000], [ 3, 0.26667, 0.00000]], [ 0.00540, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LHipPitch")
        times.append([ 1.12000, 1.80000, 2.60000])
        keys.append([ [ -1.77325, [ 3, -0.37333, 0.00000], [ 3, 0.22667, 0.00000]], [ -1.00166, [ 3, -0.22667, -0.29226], [ 3, 0.26667, 0.34384]], [ 0.13503, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LHipRoll")
        times.append([ 1.80000, 2.60000])
        keys.append([ [ -0.08433, [ 3, -0.60000, 0.00000], [ 3, 0.26667, 0.00000]], [ 0.10129, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LHipYawPitch")
        times.append([ 1.12000, 1.80000, 2.60000])
        keys.append([ [ -0.26180, [ 3, -0.37333, 0.00000], [ 3, 0.22667, 0.00000]], [ -0.23773, [ 3, -0.22667, -0.01473], [ 3, 0.26667, 0.01733]], [ -0.16563, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])

        names.append("LKneePitch")
        times.append([ 0.52000, 1.12000, 1.80000, 2.60000])
        keys.append([ [ 0.84648, [ 3, -0.17333, 0.00000], [ 3, 0.20000, 0.00000]], [ 1.42244, [ 3, -0.20000, 0.00000], [ 3, 0.22667, 0.00000]], [ 1.00780, [ 3, -0.22667, 0.23195], [ 3, 0.26667, -0.27289]], [ -0.09208, [ 3, -0.26667, 0.00000], [ 3, 0.00000, 0.00000]]])
```

Fig. 12 . Used joints' representation in choregraphe (for sitting position)

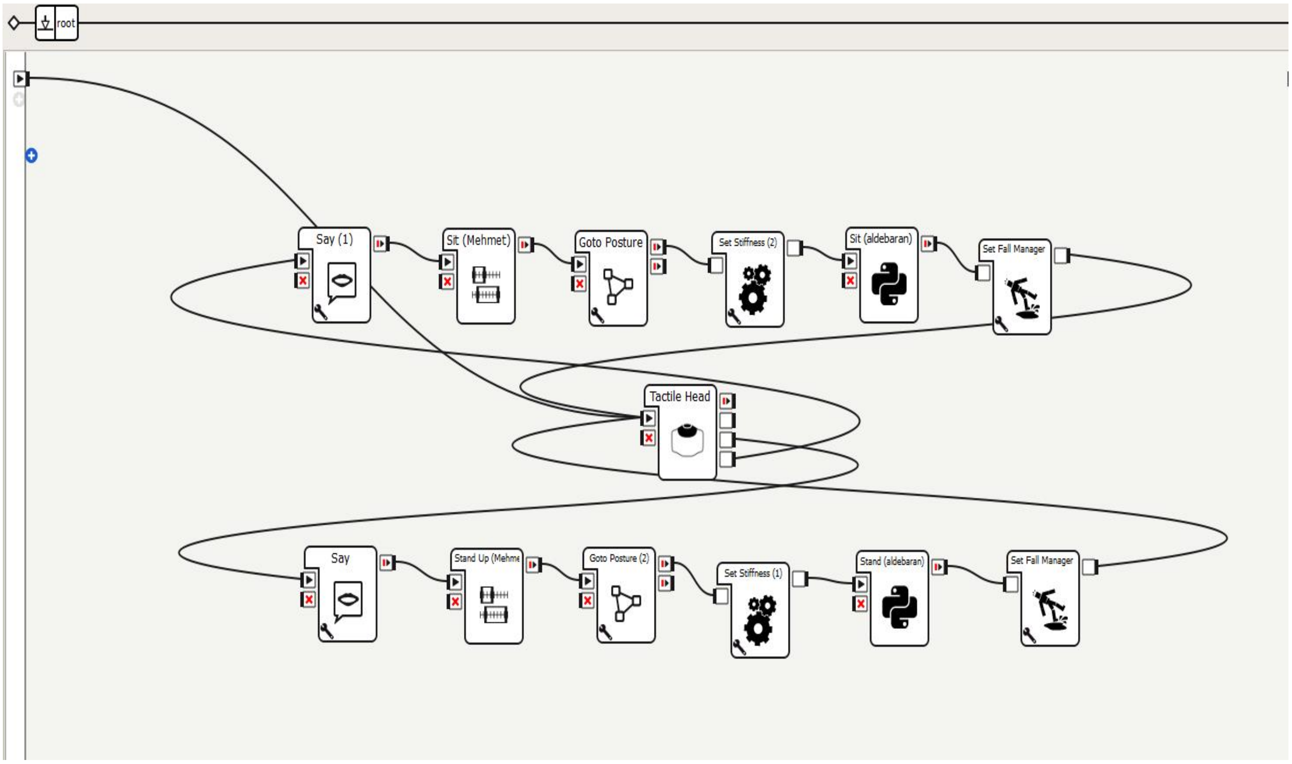


Fig. 13 . STS motion algorithm implementation in choreographe

ISSN: 2414-4592
DOI: 10.20474/jater-3.6.4

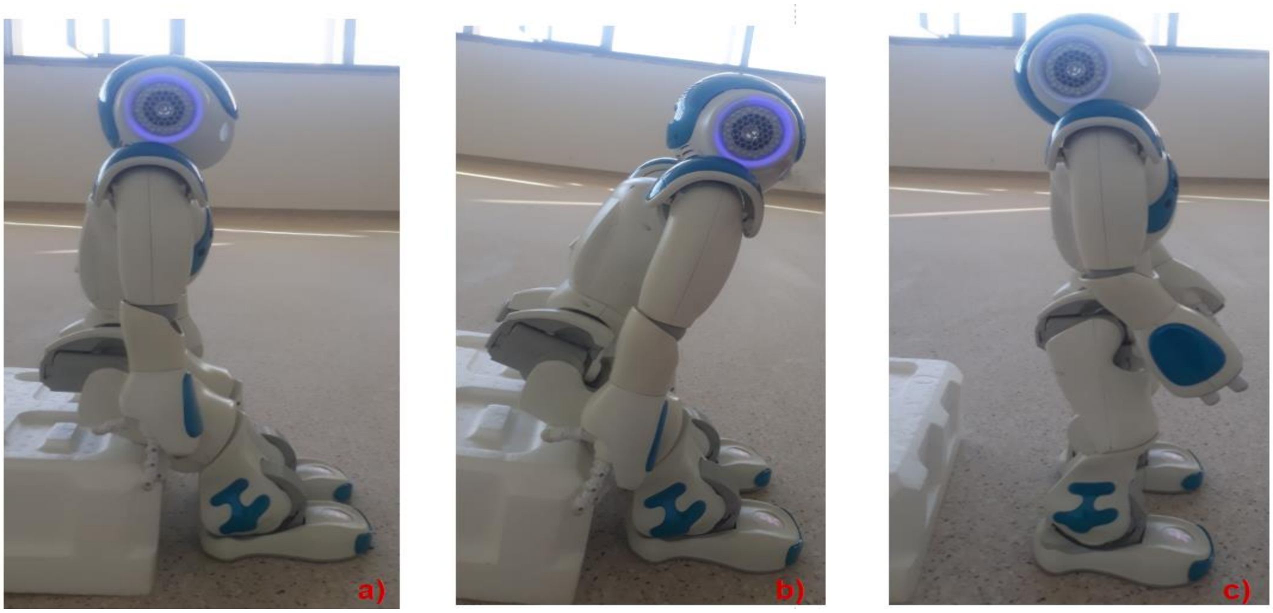


Fig. 14 . The STS motion algorithm application to NAO humanoid robot, a) NAO humanoid robot is at initial sitting position, b) NAO Humanoid robot prepares itself for standing up position, c) NAO Humanoid robot moves to the standing position

V. CONCLUSION AND FURTHER RESEARCH

In this paper, STS motion algorithm for STS task is reviewed and applied to the NAO humanoid robot both in the simulation and experimental environments. For achieving STS tasks, initially COM position of the robot is determined and then desired joint angles for the hip, knee, and ankle joints are specified. Finally, the obtained joint angles and corresponding whole body movements for the STS tasks are evaluated. As can be understood from the results, the proposed technique is able to control the STS motion of the robot. Furthermore, applied STS motion algorithm is able to transfer the HAT CoM into the defined support polygon area to increase the stability of whole movement. Even though the proposed algorithm can be modified and implemented to all humanoid robots, in this paper, it has been applied to NAO humanoid robot. This research will be extended with the object recognition and selection algorithms for various sophisticated Human-Robot Interaction cases in the near future.

ACKNOWLEDGMENT

We would like to thank TUBITAK as this work is supported by TUBITAK with project 215E047.

REFERENCES

[1] O. Chuy, Y. Hirat, Z. Wang and K. Kosuge, "Approach in assisting a sit-to-stand movement using robotic walking support system," in *Proceeding of International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[2] K. A. Strausser and H. Kazerooni, "The development and testing of a human machine interface for a mobile medical exoskeleton," in *Proceedings of International Conference on Intelligent Robots and Systems*, San Francisco, CA, 2011.

[3] M. Mistry, A. Murai, K. Yamane and J. Hodgins, "Sit-to-stand task on a humanoid robot from human demonstration," in *Proceedings of International Conference on Humanoid Robots (Humanoids)*, Nashville, TN, 2010.

[4] X. Gu and D. H. Ballard, "Motor synergies for coordinated movements in humanoids," in *Proceedings of International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[5] S. Pchelkin, A. Shiriaev, L. Freidovich, U. Mettin, S. Gusev and W. Kwon, "Natural sit-down and chair-rise motions for a humanoid robot," in *Proceedings of 49th IEEE Conference on Decision and Control*, Atlanta, GA, 2010.

[6] M. Sugisaka, "A control method for soft robots based on artificial musles," in *Proceedings of 4th IEEE International Conference on Mechatronics*, Harbin, China, 2007.

[7] P. O. Riley, R. Popat and D. E. Krebs, "Momentum analysis of sitback failures in sit-to-stand trials," In *Proceedings of IEEE 17th Annual Conference on Engineering in Medicine and Biology Society*, Montreal, Canada, 1995.

[8] K. Punaiyah and H. Singh "Biped robot for walking and turning motion using raspberry Pi and Arduino," *International Journal of Technology and Engineering Studies*, vol. 3, no. 2, pp. 49-58, 2017. **DOI:** 10.20469/ijtes.3.40002-2

[9] M. B. Bahar, M. F. Miskon, N. A. Bakar, F. Ali and A. Z. Shukor, "STS motion control using humanoid robot," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 8, no. 1, pp. 95-108, 2014. **DOI:** 10.19026/rjaset.8.945

[10] A. B. Schultz, N. B. Alexander and J. A. Ashton-Miller, "Biomechanical analyses of rising from a chair," *Journal of Biomechanics*, vol. 25, no. 12, pp. 1383-1391, 1992. **DOI:** 10.1016/0021-9290(92)90052-3

[11] Y. C. Pai, B. J. Naughton, R. W. Chang and M. W. Rogers, "Control of body centre of mass momentum during sit-to-stand among young and elderly adults," *Gait & Posture*, vol. 2, no. 2, pp. 109-116, 1994. **DOI:** 10.1016/0966-6362(94)90100-7

[12] T. Bajd, A. Kralj and R. Turk, "Standing-up of a healthy subject and a paraplegic patient," *Journal of Biomechanics*, vol. 15, no. 1, pp. 1-10, 1982. **DOI:** 10.1016/0021-9290(82)90029-x

[13] S. S. Coghlin and B. J. McFadyen, "Transfer strategies used to rise from a chair in normal and low back pain subjects," *Clinical Biomechanics*, vol. 9, no. 2, pp. 85-92, 1994. **DOI:** 10.1016/0268-0033(94)90029-9

[14] S. Hesse, M. Schauer, M. Malezic, M. Jahnke and K. H. Mauritz, "Quantitative analysis of rising from a chair in healthy and hemiparetic subjects," *Scandinavian Journal of Rehabilitation Medicine*, vol. 26, no. 3, pp. 161-166, 1994.

[15] B. Bahar, M. F. Miskon, N. A. Bakar, A. Z. Shukor and F. Ali, "Path generation of sit to stand motion using humanoid robot," *Australian Journal of Basic Applied Sciences*, vol. 8, pp. 168-182, 2014.

[16] R. Prinz, S. Neville and N. J. Livingston, "Development of a fuzzy-based sit-to-stand controller," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, Vancouver, Canada, 2007.

[17] K. Kondak and G. Hommel, "Control and online computation of stable movement for biped robots," in *Proceedings of International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003.

[18] G. Rasool, A. M. Mughal and K. Iqbal, " Fuzzy biomechanical sit-to-stand movement with physiological feedback latencies," in *Proceedings of IEEE International Conference on Systems Man and Cybernetics*, Istanbul, Turkey, 2010.

[19] P. Little, G. Lewith, F. Webley, M. Evans, A. Beattie, K. Middleton, J. Barnett, K. Ballard, F. Oxford, P. Smith and L. Yardley, "Randomised controlled trial of Alexander Technique lessons, Exercise, and Massage (ATEAM) for chronic and recurrent back pain," *British Medical Journal*, vol. 337, pp. 1-8, 2008. **DOI:** 10.1136/bmj.a884

[20] Aldebaran-Robotics, "Contact and tactile sensors," 2012. [Online]. Available: goo.gl/V2wn9W

[21] Aldebaran-Robotics, "NAO motors," 2012. [Online]. Available: goo.gl/eu7WgJ

— This article does not have any appendix. —