



HAL
open science

Nut fastening with a humanoid robot

Kai Pfeiffer, Adrien Escande, Abderrahmane Kheddar

► **To cite this version:**

Kai Pfeiffer, Adrien Escande, Abderrahmane Kheddar. Nut fastening with a humanoid robot. IROS: Intelligent Robots and Systems, Sep 2017, Vancouver, BC, Canada. pp.6142-6148, 10.1109/IROS.2017.8206515 . hal-02189053

HAL Id: hal-02189053

<https://hal.science/hal-02189053v1>

Submitted on 19 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Nut Fastening with a Humanoid Robot

Kai Pfeiffer^{1,2}, Adrien Escande¹ and Abderrahmane Kheddar^{1,2}

Abstract—We study the HRP-2Kai humanoid robot’s ability to conduct the precise industrial task of fastening bolts in aircraft production. Our contribution stands mainly in high integration of different modules within the whole-body Quadratic Programming-based controller that has not been yet confronted to tasks demanding high precision in the execution and tuning. This includes the use of a robust visual servoing algorithm which allows the robot to move autonomously to a desired target and the design of specific tasks and estimators: a learning and admittance control that enables the robot to interact smoothly with its environment, and a fast and safe method to autonomously detect correct tool on nut insertion. We then show that the controller indeed enables our humanoid robot to achieve such a high precision task.

I. INTRODUCTION

Aircraft assembly is conducted onsite due to the amount of large scale and individual operations especially during the fusion of the main body consisting of the fuselage, wings and empennage, and the installation of the inner equipment. Most of this assembly consists of many repetitive tasks requiring high precision over large structures like drilling, bolt and nut insertion, the consecutive fastening, riveting, welding and cable laying. All these tasks are subjected to high quality standards in order to guarantee the aircraft’s safety. These tasks being conducted by human workers inherit the danger of errors due to concentration loss caused by the high repetitiveness and physical exhaustion when working over head or in difficultly accessible places. Those tasks have low added value (in that the know-how of the human operator is not essential) but errors in their execution are costly to repair.

In order to overcome these issues automation represents a viable option. While classical mounted or servo tracked industrial robots are already widely used in many industries (e.g. automotive industry), such robots are adversely affected by their immobility and the inaccessibility especially of the inner of the aircraft and in general cannot be applied universally for tasks of different nature. With Airbus, we are studying the feasibility of using humanoid robots as a vector of automation. It has been shown that those robots are able to execute tasks which are mostly done by humans in today’s industrial world. Their anthropomorphic structure and diverse sensory feedback enables walking on uneven terrains [1], climbing ladders [2], driving cars [3] and many other tasks as seen e.g. in the DARPA robotics challenge.

As one of the steps to the bigger goal of humanoid robots working autonomously in aircraft assembly we have

identified the common process of fastening nuts onto bolts. Their robustness make bolt connections explicitly applicable in aircraft production where high safety is required which cannot be ensured for example with glued contacts. Therefore this kind of joining procedure is conducted uncountable times in aircraft assembly and could see a significant leap forwards with automation in terms of working speed and quality when comparing with human conductors.

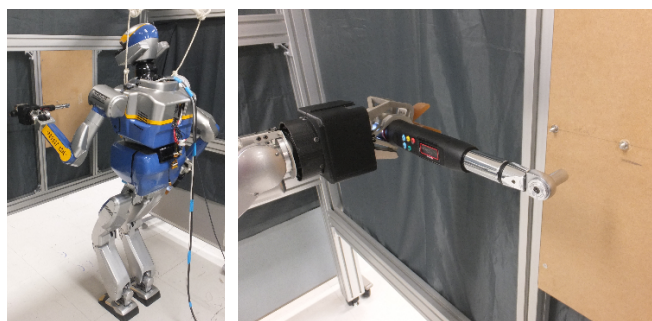


Fig. 1. Full body shot of the HRP-2Kai in front of the wall and close up of the wrench tool with tool-adaptor and the bolt nut connections.

From a scientific and technological point of view, this use-case is the occasion to apply our Quadratic Programming (QP)-based controller [2] in a high-precision industrial scenario, with a type of robot that is not (yet) designed with such an application in mind. Over the past decade, we and many others have applied variants of the QP-based control approach to various simple tasks and multi-contact scenarios (see [4], [5], [6], [7], [8], [2] to name but a few), but the control of a full-scale humanoid robot to perform high precision manipulation tasks is a new research direction.

This paper is a first step in this direction. It highlights the efforts to be done, and proposes a solution to perform nut fastening by introducing several dedicated tasks in our QP formulation. We first describe the problem we are solving (Sec. II), then introduce our overall control framework and the specific tasks (Sec. III), that we detail in further sections. In Sec. IV the trajectory design for the fastening movement is described. It is executed with an admittance control which allows safe interaction between the robot and its environment, together with trajectory learning to reduce the unwanted reaction forces occurring. Before executing the fastening motion, we need to check quickly and reliably that the tool is correctly inserted. The proposed method is described in Sec. V. The experimental validation of the described methods is then presented (Sec. VI), before concluding (Sec. VII).

This research is supported in part by the CNRS-AIST-AIRBUS Joint Research Program, and the EU H2020 COMANOID project.

¹CNRS-AIST JRL (Joint Robotic Laboratory) UMI3218/RL, Tsukuba, Japan

²CNRS-UM2 LIRMM UMR 5506, 34095 Montpellier Cedex 5, France

II. PROBLEM DESCRIPTION

We study the problem of fastening a nut onto a bolt, with a humanoid robot using a wrench (see Fig. 1). We use a commercially available electronic wrench, whose reference was given to us by Airbus so that it closely matches typical wrenches found in Airbus factories. In particular it can be configured so that reverse motion does not unfasten the nut.

The used nut and bolt are also classical commercial ones. In the experiments of section VI, they have diameters of 10mm and 5mm, respectively. Bolts are fixed to a mockup we are building upon Airbus specifications to mimic situations found in an airplane assembly.

Since the combination of currently available visual-tracking libraries and the used robot camera is not powerful enough to reliably and accurately track small sized and reflecting metallic nut, we ease the visual feedback using markers.

We also make the following assumptions:

- The robot is placed close enough to the nut so that it can perform the task without changing contacts (however the position of the robot relatively to the nut is unknown).
- The tool is already placed in the robot's hand.
- The nut is already inserted onto the bolt.
- The position of the points to be visually tracked with respect to their markers is known.

Even with these simplification, the fastening task presents several challenges. First, the tool-tip must be inserted correctly onto the nut. This requires a sub-millimeter and sub-degree positioning, but (i) the relative position between the nut and the tool-tip is not given that precisely by the vision, and (ii) the rotation of the hexagonal nut around the bolt is only known to lay within a range of 60°. Even if the vision system was perfect, the tool-tip is occluding the view when nearing the nut. As a consequence, the insertion has to be executed based only on force sensing. Secondly, the placement of the tool in the robot hand is not perfectly known, and the robot and the controller do not necessarily have the required precision to perform the fastening motion perfectly (errors in the controller model, flexibility at the ankle, etc.): in preliminary tests performing a fastening motion without nut insertion we measured the deviation of the tool-tip from the desired fixed position to lay within 1-2cm. Once the nut is actually inserted this will generate internal efforts that need to be reduced.

The fastening procedure we consider is:

- 1) Positioning of the tool in front of the nut within short distance by visual servoing;
- 2) Contacting the nut;
- 3) Inserting the nut into the tool by a test movement;
- 4) Check if the nut is inserted correctly;
- 5) Fastening until the desired torque level is reached.

III. MULTI-SENSORY QP CONTROLLER

In this section, we define our QP-based controller. The goal of the controller is to compute at each time step dt an

acceleration \ddot{q} , where q is the configuration of the robot, so as to fulfill (at best) a set of objectives. This acceleration must be physically feasible, *i.e.* there exist a set of contact forces f and joint torques τ such that the equation of motion $M(q)\ddot{q} + n(q, \dot{q}) = L\tau + J_c^T f$ is satisfied, the forces are within their Coulomb friction cones and the joint torques are within their limits. Here, M denotes the inertia matrix of the robot, n gathers the Coriolis and gravity effects, $L = [0 \ I]^T$ is a selection matrix accounting for the underactuation of the robot base, and J_c is the Jacobian matrix of the contact points.

Geometric objectives are defined as functions $e_i(q)$ that we want to regulate to zero or maintain above zero. We name the resulting constraints $e_i(q) = 0$ and $e_i(q) \geq 0$, *tasks*. Typical tasks includes

- Posture task: $e_{\text{posture}} = q - q^d = 0$
- Bound tasks: $e_{\text{sup}} = \bar{q} - q \geq 0$ and $e_{\text{inf}} = q - \underline{q} \geq 0$
- CoM task: $e_{\text{CoM}} = \text{CoM}(q) - \text{CoM}^d = 0$
- Body i position task: $e_{\text{pos}} = x_i(q) - x_i^d = 0$
- Body i orientation task: $e_{\text{or}} = \log(R_i(q)^T R_i^d) = 0$
- Collision avoidance $e_{\text{col}} = \delta_{i,j}(q) \geq 0$

where the superscript d denotes a desired value specified by the user, \underline{q} and \bar{q} are the bounds of q , x_i and r_i are the position and orientation of the i -th robot body, \log is the logarithm over $\text{SO}(3)$ (see [9]), and $\delta_{i,j}$ denotes the distance between body i and j (one of which can be a part of the environment). The regulation of a task e_i is made by specifying the behavior of its acceleration \ddot{e}_i^* , that is $\ddot{e}_i = \ddot{e}_i^*$ or $\ddot{e}_i \geq \ddot{e}_i^*$. Noting that $\ddot{e}_i = J_i(q)\ddot{q} + \dot{J}_i(q)\dot{q}$, with J_i the Jacobian matrix of e_i , this specification is a linear equality or inequality constraint in \ddot{q} . In the equality case, the desired acceleration is usually taken as a critically damped PD with gain k_i : $\ddot{e}_i^* = -2\sqrt{k_i}\dot{e}_i - k_i e_i$. In the inequality case, we take an acceleration damper $\ddot{e}_i^* = \frac{\xi}{dt} \frac{e_i - k_s}{k_i - k_s} - \frac{\dot{e}_i}{dt}$, with parameters ξ , k_s and k_i (see [2] for more details).

Some linearized tasks may not be feasible and some others may be conflicting, so that we might want to solve them at best, in a least-square sense. We define the quadratic cost

$$c_i(q, \dot{q}, \ddot{q}) = \frac{1}{2} \|J_i \ddot{q} + \dot{J}_i \dot{q} - \ddot{e}_i\|^2$$

corresponding to a task $e_i = 0$. We suppose that inequality tasks are feasible all together.

This leads to the following quadratic program:

$$\begin{aligned} \min_{\ddot{q}, \tau, f} \quad & \sum_{i \in O} \omega_i c_i(q, \dot{q}, \ddot{q}) + \omega_f \|f\|^2 \\ \text{s.t.} \quad & M(q)\ddot{q} + n(q, \dot{q}) = L\tau + J_c^T f \\ & \underline{\tau} \leq \tau \leq \bar{\tau}, \quad Cf \geq 0 \\ & J_i \ddot{q} + \dot{J}_i \dot{q} = \ddot{e}_i^* \forall i \in E \\ & J_i \ddot{q} + \dot{J}_i \dot{q} \geq \ddot{e}_i^* \forall i \in I \end{aligned}$$

with E , O and I denoting respectively the tasks included as equality constraints (usually the contact tasks), those included as objectives, and those included as inequality constraints. The constraint $Cf \geq 0$ is the discretized version

of the inclusion of f in the Coulomb friction cones. $\underline{\tau}$ and $\bar{\tau}$ are the bounds on τ . The errors c_i of the different tasks i are weighted against each other by the weight ω_i depending on their relative importance and desired accuracy. The damping term with weight ω_f ensures the uniqueness of the solution.

By taking into account the shape of L and pre-solving in τ , the above QP formulation is solved by the LSSOL solver [10] in about 1ms for the case of two contacts [2]. The resulting \ddot{q} is integrated twice to be sent as a command for the robot.

This control framework is extremely flexible and extensible. A visual servoing task can be added in a similar manner as the geometry tasks described above. For this we follow the approach of [11].

The fastening motion is based on a body position and a body orientation tasks for the hand of the robot. Its particular design consists in a specific computation of the target position and orientation x^d and R^d that we describe in the next section.

IV. FASTENING TRAJECTORY

A. Basic trajectory

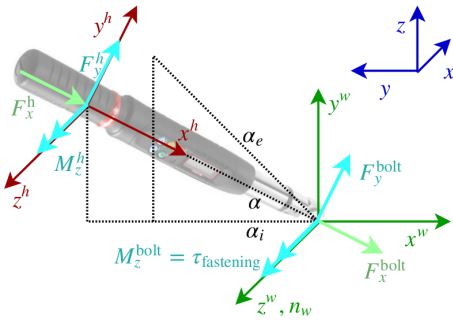


Fig. 2. Free body diagram of the wrench tool: world (w , dark blue), wall (w , dark green) and hand (h , red) reference systems; torques and forces which are dependent of each other are colored the same: $F_x^h = -F_x^{\text{bolt}}$, $F_y^h = -F_y^{\text{bolt}}$, $M_z^h = -\tau_{\text{fastening}} - L_{\text{wrench}} F_y^{\text{bolt}}$, ($\tau_{\text{fastening}} = -M_z^h + L_{\text{wrench}} F_y^h$)

The fastening trajectory is designed as a circle of radius L_{wrench} (the length between hand reference frame and the tool-tip) around the rotation center / bolt position x_{bolt} :

$$x_{\text{basic}} = x_{\text{bolt}} + R_w \begin{bmatrix} -L_{\text{wrench}} \cos(\alpha) \\ L_{\text{wrench}} \sin(\alpha) \\ 0 \end{bmatrix} \quad R_{\text{basic}} = R^z(\alpha) R_w^T. \quad (1)$$

R_w is the orientation of a frame attached to the bolt with the z axis aligned with the bolt, and $R^z(\alpha)$ is the rotation of angle α around the z^w axis. x_{basic} and ϕ_{basic} are the desired position and orientation of the hand frame along this trajectory. α is designed as a minimum jerk parametrization from the initial angle α_i to the end angle α_e in N seconds.

$$\alpha(k) = \alpha_i + (\alpha_e - \alpha_i) \left(10 \left(\frac{k}{N} \right)^3 - 15 \left(\frac{k}{N} \right)^4 + 6 \left(\frac{k}{N} \right)^5 \right)$$

With this parametrization, we can generate an alternate sequence of clockwise (cw) and counterclockwise (ccw)

motions by switching α_i and α_e . Our tool can be configured so that cw motion fasten the nut whereas ccw ones leave it unmoved. When the target fastening torque is attained, we stop the motion at the current angle.

B. Admittance control

Since every QP objective is only solved at best in the least-squares sense and the robot state is not perfectly known a certain amount of inaccuracy of the robot movement remains. Additionally, the interaction between the tool and the nut is unknown. Therefore, a compliant behavior of the end effector with its environment will allow safe interaction. A classical admittance law in the 3-D task space is implemented which connects the external forces F_{ext} and torques T_{ext} and the corresponding translational and rotational displacements x and ϕ as follows (the quantities with subscript h are expressed in the force sensor / hand frame):

$$\Delta F_{\text{ext}}^h = M \Delta \ddot{x}^h + D \Delta \dot{x}^h + K \Delta x^h \quad (2)$$

$$\Delta T_{\text{ext}}^h = J \Delta \ddot{\phi}^h + L \Delta \dot{\phi}^h + C \Delta \phi^h \quad (3)$$

$$\Delta F_{\text{ext}}^h = F_{\text{ext}}^{\text{measured},h} - F_{\text{ext},d}^h \quad (4)$$

$$\Delta T_{\text{ext}}^h = T_{\text{ext}}^{\text{measured},h} - T_{\text{ext},d}^h \quad (5)$$

Since all the gain matrices are chosen to be diagonal this decoupled second order ODE system can be solved for Δx^h and $\Delta \phi^h$ by the finite difference method. The control output is then composed as follows:

$$x_{\text{control}}^* = x_{\text{basic}} + R_h \Delta x^h \quad (6)$$

$$R_{\text{control}}^* = R(\Delta \phi^h) R_{\text{basic}}. \quad (7)$$

R_h is the rotation matrix from the force sensor to the world reference system and $R(\Delta \phi^h)$ describes the 3D rotation by ϕ^h which is composed of the angles calculated from the ODE.

C. Trajectory learning control

During the fastening movement rather fast wrench characteristics were observed repeatedly for every turn. The admittance control which acts as a low-pass filter and therefore possesses a phase is not perfectly suitable for tackling such fast error behaviors. Applying a phase-free learned trajectory Ψ constitutes a simple solution (Fig. 3) [12]. At every turning movement i the measured wrench error $\Delta \Gamma = [\Delta F_{\text{ext}}^h; \Delta T_{\text{ext}}^h]$ is recorded; then, in an offline process, parallel to the control, it is zero-phase low-pass filtered and finally applied with a certain learning gain at the following cw or ccw motions. For physical consistency with the admittance control a low-pass filter designed as a mass-damper-spring system with gains $m = 1.0\text{kg}$, $d = 1.0\text{kg/s}$ and $k = 1.0\text{kg/s}^2$ is used. Additionally, in order to pertain repeatability of the wrench characteristics and in order to reset the initial conditions of the system after every trial, the learned wrenches are convoluted by a Tukey window τ [13] which becomes zero at both ends $t = 0\text{s}$ and $t = N\text{s}$:

$$\Psi_{i+1} = \Psi_i + \tau k_{\text{learning}} \Delta \Gamma_i^{0\varphi} \quad (8)$$

The overall control then writes as

$$\begin{bmatrix} x^d \\ R^d \end{bmatrix} = \begin{bmatrix} x_{\text{control}}^* + R_h \Psi_{\text{lin}} \\ R(\Psi_{\text{rot}}) R_{\text{control}}^* \end{bmatrix}$$

and is fed to the QP-controller as desired trajectory of the end-effector. Thereby $R(\Psi_{\text{rot}})$ describes the 3D rotation by Ψ_{rot} which is composed of the learned angles.

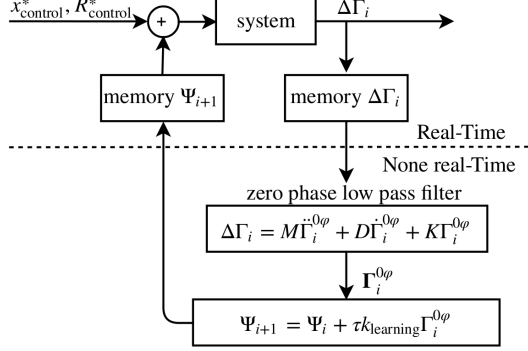


Fig. 3. The trajectory learning control diagram: The system is unknown and outlines the robot's dynamics and its interaction with the environment. Its input are the desired trajectories for the end-effector x^d and R^d and its output is the measured wrench error $\Delta\Gamma$.

V. DETECTION OF CORRECT NUT IN TOOL INSERTION

We based the insertion detection on the following property: if the tool is correctly positioned any movement in the wall plane will lead to a reaction force F pointing to the bolt. Mathematically this can be expressed by

$$\frac{x_{y^w} - x_{y^w}^{\text{bolt}}}{x_{x^w} - x_{x^w}^{\text{bolt}}} = \frac{F_{y^w}}{F_{x^w}} \text{ for } |F_{x^w}| \geq |F_{y^w}|, |F_{y^w}| > 0. \quad (9)$$

which expresses the directional alignment of the reaction force F and the displacement vector $x^{\text{bolt}} - x$ between the current tool position and the yet unknown center of the bolt x_{bolt} (see Fig. 4). The x^w - and y^w - axis define the plane of the wall from which the bolt is perpendicularly sticking out. The detection method consists of moving the tool-tip (exploration movement) and verifying that this is indeed the case. Depending on which force component is larger and in

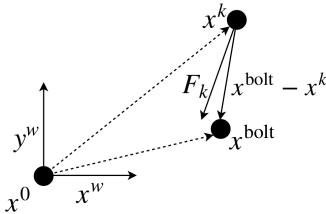


Fig. 4. Vectors F_k and $x_{\text{bolt}} - x_k$ at instant k used for the calculation of the force field characteristic FFC. x^0 is the current tool position at $k = 0$ in the world reference system. It is the origin of the wall reference system where the bolt position x_{bolt} is regressed.

order to avoid singularities the inverse of the above equation 9 can be used:

$$\frac{x_{x^w} - x_{x^w}^{\text{bolt}}}{x_{y^w} - x_{y^w}^{\text{bolt}}} = \frac{F_{x^w}}{F_{y^w}} \text{ for } |F_{x^w}| < |F_{y^w}|. \quad (10)$$

Reformulating (9) and (10) into vector form and sampling the equations over time ($k = 1 \dots n$) yields a linear regression problem

$$\begin{bmatrix} \frac{F_{y^w}^k}{F_{x^w}^k} & -1 \end{bmatrix} \begin{bmatrix} x_{x^w}^{\text{bolt}} \\ x_{y^w}^{\text{bolt}} \end{bmatrix} = \frac{F_{y^w}^k}{F_{x^w}^k} x_{x^w}^k - x_{y^w}^k \text{ for } |F_{x^w}^k| \geq |F_{y^w}^k| > 0$$

$$\begin{bmatrix} 1 & -\frac{F_{x^w}^k}{F_{y^w}^k} \end{bmatrix} \begin{bmatrix} x_{x^w}^{\text{bolt}} \\ x_{y^w}^{\text{bolt}} \end{bmatrix} = x_{x^w}^k - \frac{F_{x^w}^k}{F_{y^w}^k} x_{y^w}^k \text{ for } |F_{x^w}^k| < |F_{y^w}^k|$$

which can be solved for x_{bolt} by least squares.

This formulation leads to a well-conditioned regressor with condition numbers usually ranging from $1.0 < \kappa < 5.0$ and thus makes it a reliable decision variable. For the 2-D case the composed regressor is only half the size of the regressor used in [14].

The information of whether the force is pointing to or pointing away from the bolt center is lost when formulating equations (9) and (10). This information has to be restored in order to learn the structure of the force field. The assumption is that in the case of a correct tool insertion the force field is attracting and all the forces point towards the bolt center. We define the scalar force field characteristic FFC using the bolt center resulting from the least squares regression:

$$\text{FFC} = \frac{1}{n} \sum_{k=1}^n \frac{x^{\text{bolt}} - x^k}{\|x^{\text{bolt}} - x^k\|} \cdot \frac{F^k}{\|F^k\|}. \quad (11)$$

The FFC can take values between -1.0 (source, totally repelling) and 1.0 (sink, totally attracting). A good threshold for a correct insertion would be $\text{FFC} \geq 0.8$ when accounting for an erroneous robot state, noise afflicted and offset force sensors and occurring friction when moving along the exploration trajectory.

However, this value alone cannot determine a correct insertion by itself but requires further data examination:

- The *condition number* κ of the regressor can give a good hint on whether the exploration movement and the corresponding forces contain enough information for a good identification. For a correct insertion the FFC will always have a high value but this applies also for other cases for example if the force sensor has a constant offset but otherwise no external forces occur. However, the condition number will be high for these cases and will indicate that the tool insertion was not successful. In this work a threshold of 2.0 was chosen.
- A *drifting exploration movement* implies that the tool is not correctly inserted. For a correct insertion it is assumed that the movement lays within $\|x_{\text{initial}} - x_k\| \leq 0.015\text{m}$ when considering both robot elasticity and clearance between the tool and the nut.
- In order to ignore force sensor background noise only forces with a norm above the threshold of 1.0N are considered for the least squares regression. Also, an insertion is only deemed successful if at least 50% of the recorded points lay above the threshold.

A. Exploration movement

Instead of moving along a predefined trajectory the robot should be able to explore the force field F_k au-

tonomously [15] [16]. A good motion should lead to a

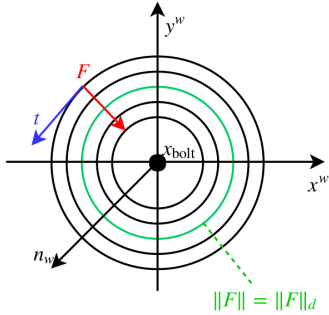


Fig. 5. Force field around the bolt center: At the bolt center itself $\|F\| = 0\text{N}$ is assumed. Ideally the robot end-effector would move along the green contour line where a desired constant force norm is given.

well-conditioned regressor and result in only small reaction forces when contacting the environment. The first component achieves the exploration character by moving along the tangent of the force field

$$t = n_w \times F \quad (12)$$

with n_w being the normal of the wall. For a perfectly circular 2-D force field this leads to a circulating movement around the center of the force field. An additional component is necessary to keep the robot end-effector on a single contour line with a desired constant force norm. This can be achieved by controlling the error of the force norm

$$\epsilon_F = \|F\| - \|F\|_d. \quad (13)$$

along the direction of the gradient of the force field $F/\|F\|$ (Fig. 5). The superposition of the two components results in the overall control output

$$c = h_{\text{tangent}}t + h_{\text{force}}\epsilon_F \frac{F}{\|F\|} \quad (14)$$

with h being the respective control gains ($h_{\text{tangent}} \approx 3h_{\text{force}}$). This control output is then added to the current end effector position at every time instance k :

$$x_{k+1}^d = x_k^{\text{measured}} + c_k \quad (15)$$

VI. EXPERIMENTAL VALIDATION

For the experiments (Fig. 6) Kawada HRP-2Kai humanoid robot with 36 DoF's was used. The robot's grippers are mounted on 3-D wrench sensors which measure external forces and torques. All end-effector movements are assumed to be slow such that a purely static gravity compensation is sufficient. For the fastening we use a FACOM E.306-30D electronic torque wrench with a weight of 1.52kg which can measure up to 30Nm. The hexagonal nut is fastened onto a M5 bolt sticking 1cm out of a wooden plate which is fixed perpendicularly in a constrained airplane mockup of dimensions $2 \times 2 \times 2\text{m}^3$. The robot fastens the nut twice with a pre-set number of fastening turns which resulted in reaction forces of the left end-effector as seen in Fig. 8. At one point of the insertion processes the robot misses the nut deliberately which can be seen in Fig. 7. The robot possesses

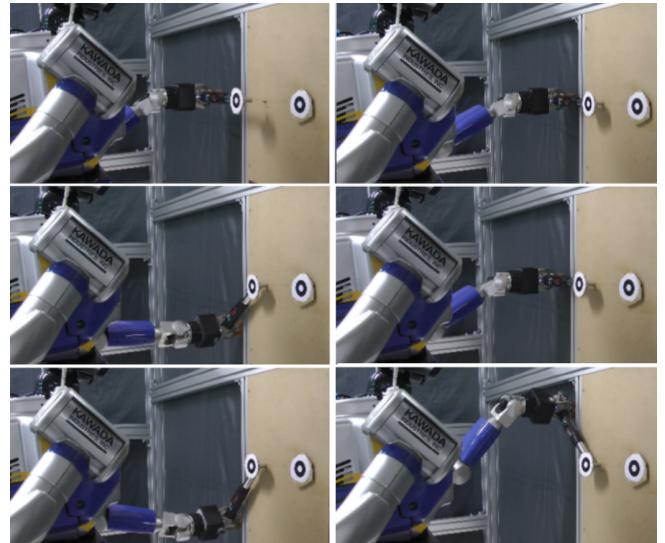


Fig. 6. Different robot steps from left to right and top to bottom: wall approach and wall contact, nut insertion, insertion check, nut fastening during cw and ccw movement

a Xtion RGB-D camera. The marker tracking is done by the open source software whycon [17] with the 3D marker positions in image frame as output.

A. Visual servoing and tool insertion

For accurate knowledge of the relative position between the bolt and the tool whycon markers are placed on each the tool and the wooden plate. The head is pointing to a hand-picked position such that the bolt marker is visible in the right half of the captured image. The tool is placed into the left hand and this hand is then steered approximately into the field-of-view of the camera. From this position the visual servoing starts and first steers the end-effector in front of the nut. The nut is then approached until a contact is detected which is indicated by an exceeded force threshold in wall normal direction. At this stage the nut and the tool are most likely misaligned rotationally. Therefore the robot proceeds to execute a test fastening movement while pressing normally against the nut. This leads to a full insertion usually within two turns with compliance applied to F_x^h and M_z^h .

B. Insertion detection

Since the tool and nut connection might be visually occluded by the robot hand a tactile check of the correct tool insertion seems to be the most practical. With the method described in section V both the cases of a inserted and non-inserted tool can be detected reliably.

For the case of a correct insertion of the tool onto the nut (Fig. 7, left) we can see the exploration movement circulating around the force field center. The analysis of the 10s and 0.054m long exploration movement with $\text{FFC} = 0.90$, $\kappa = 1.10$, 95% of valid data points with $\|F\| \geq 1.0\text{N}$ and with a peak reaction force of $\|F\|_{\text{max}} = 3.33\text{N}$ correctly signals the tool insertion. The clearance between the wrench tool

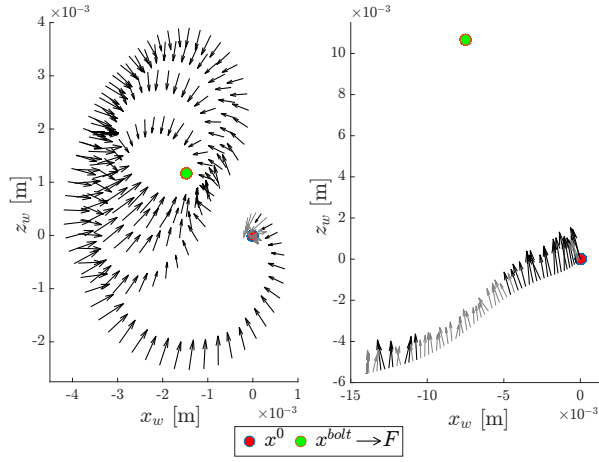


Fig. 7. Force field detected for inserted (left) and non-inserted tool (right); forces with $\|F\| < 1\text{N}$ are coloured gray.

and the tool-adaptor and between the tool-adaptor and the nut explains the low forces occurring at the beginning of the exploration.

For a non-inserted tool (Fig. 7, right) the end effector clearly drifts away from the starting point and stops after the threshold $\|x_{\text{initial}} - x^k\| \leq 0.015\text{m}$ is surpassed. The analysis of the 3.1s long exploration movement resulted in $\text{FFC} = 0.96$, $\kappa = 4.93$, 16% of valid data points with $\|F\| \geq 1.0\text{N}$ and a peak reaction force of $\|F\|_{\text{max}} = 1.68\text{N}$ and therefore indicates correctly that the tool is not inserted. The recorded forces all point towards a similar direction which causes the condition number to lay above the threshold. This is a more reliable measure to detect incorrect insertion than the FFC which by itself would indicate a correct insertion.

The corresponding gains are $h_{\text{tangent}} = 0.006$ and $h_{\text{force}} = 0.00225$ which leads to a quasi static exploration movement with a speed of $\simeq 0.54\text{cm/s}$.

C. Fastening process

The circular fastening trajectory is set to range from $\alpha_e^{\text{down}} = -45^\circ$ to $\alpha_e^{\text{up}} = 45^\circ$ and is executed in a perpendicular plane approximately 0.9m in front of the robot. This enables a motion in the feasible workspace of the robot. For a high accuracy the duration of the cw and ccw trajectories is set to 10s. In order to enable a safe interaction the following DoF's of the end-effector are subject to the admittance control: x^h, z^h . The desired force $F_{z,d}^h = 3.5\text{N}$ results in the robot pressing the end effector against the wall and preventing the tool from losing touch with the nut during the execution of the fastening movement. The two other components in radial direction along the tool axis $F_{x,d}^h = 0\text{N}$ and the torque about the bolt axis $M_{z,d}^h = 0\text{Nm}$ help to keep the reaction forces on the nut low. The admittance gains are chosen as follows in order to create a slow but compliant

behavior:

$$M_{x,z} = 2000\text{kg} \quad D_{x,z} = 1600 \frac{\text{kg}}{\text{s}} \quad K_{x,z} = 20 \frac{\text{kg}}{\text{s}^2} \quad (16)$$

$$J_z = 800\text{kg} \quad L_z = 600 \frac{\text{kg}}{\text{s}} \quad C_z = 13 \frac{\text{kg}}{\text{s}^2}. \quad (17)$$

With only this control applied especially the cw movements are marked by high reaction force peaks up to 10N for F_x^h and F_z^h . These disturbance characteristics are learned such that during the succeeding respective cw and ccw movement knowledge about the wrench error to occur is available. This phase-free trajectory adaptation reduces the occurring high error peaks by 50% and leads to reaction forces under 5N for F_x^h and F_z^h respectively (Fig. 8).

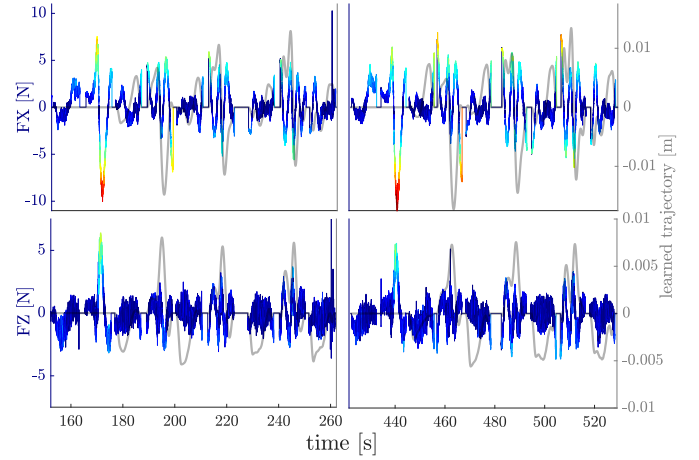


Fig. 8. Force deviations F_X and F_Z from the desired values $F_{x,d}^h = 0\text{N}$ and $F_{z,d}^h = 3.5\text{N}$ (in color) and applied learned trajectory (in gray) for the two fastening processes. The force error peaks are decreasing with time as the trajectory is refined by the learning control.

D. Process Reliability

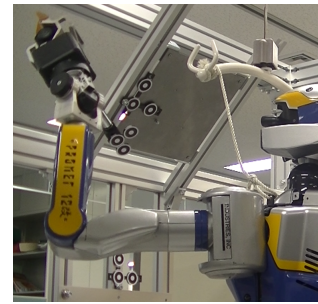


Fig. 9. HRP2-Kai fastening the nut on the upper inclined wall.

In order to confirm the process reliability the robot has to fasten consecutively two nuts onto two M8 bolts of which the first one is fixed on a 45° inclined and the second one on a perpendicular metal wall (Fig. 9). Three markers forming a L-shape are used to detect the wall orientations. The desired fastening torque is chosen to 7Nm while the current fastening torque is calculated by $\tau_{\text{fastening}} = -M_z^h + L_{\text{wrench}} F_y^h$. The fastening process was repeated 9 times in a row with

| | | | | |
|------------------|---|------|-----------------------------|---------------------|
| full success | 5 | 56% | $\tau_{\text{fastening}}$ | 8.14 ± 0.27 Nm |
| with some issues | 3 | 33% | duration/trial | $6:31 \pm 0:58$ min |
| failure | 1 | 11% | 1 st nut: #turns | 17.1 ± 2.6 |
| Σ | 9 | 100% | 2 nd nut: #turns | 11.8 ± 1.6 |

TABLE I
EXPERIMENTAL RESULTS

restarting the robot in between. 5 trials were fully successful while for 3 attempts some issue occurred (tool adapter fell off during tool removal from the first nut, several insertion attempts necessary for first or second nut). During one trial the insertion for the second nut was not successful at all. The fastening torque always lays above the desired value which can be explained by the fact that the robot is not able to stop its movement to zero in the very instant at which the torque threshold is exceeded. The overall trial duration has quite a large standard deviation of 58s which is explained for one by the more time consuming trials with some issue occurring. Additionally, for every trial the number of necessary fastening movements #turns differed slightly due to the degree of pre-fastening of the nuts. Due to geometric restrictions of the environment the fastening angle for the inclined wall had to be chosen smaller to $\alpha = 70^\circ$ instead of $\alpha = 90^\circ$ for the perpendicular wall.

VII. CONCLUSION

In this work we showed that with the addition of specific tasks in our control framework, the HRP-2Kai humanoid robot is able to conduct robustly the industrial task of fastening nuts in aircraft assembly. We achieved a high precision validated by low reaction forces on the nut.

This work highlights the importance of the sensors (force, vision) in the achievement of precise tasks with robots that were not designed for that purpose, a point that is quite new in the humanoid research community. It also raises the question of robust vision when it comes to both humanoid robots with their embedded cameras and objects with size of under 1cm, see the circuit-breaker case in [18]. We will look deeper into this in the future.

The motions presented here can be regarded as rather slow, and there are improvements to be made as well. However, software developments will not be able to compensate for all the imprecision of the hardware. There is no doubt that in order to establish humanoid automation in manufacturing, a new generation of humanoid robots has to be designed with these precision issues in mind. Yet, increasing the motion speed is definitely part of our future work.

One way to simplify manipulation problems such as nut fastening, and to increase the precision would be to mount directly dedicated tools on the robot, much like what is done with Kawada's Nextage robot, which can change its end-tool on the go. While this is an interesting prospect, it is not so easy to implement in factories such as Airbus' where the selection and certification of a tool is a heavy process. It thus makes sense to have a robot using tools designed for

humans, such as we did here, even if on a longer term the mounting of dedicated tools should bring more efficiency.

Finally, we did not fully use here the multi-contact capability of our controller. We expect the use of additional contacts to increase the quality of the manipulation motion, by giving more stability and rigidity.

REFERENCES

- [1] J.-Y. Kim, I.-W. Park, and J.-H. Oh, "Walking control algorithm of biped humanoid robot on uneven and inclined floor," *Journal of Intelligent and Robotic Systems*, vol. 48, no. 4, pp. 457–484, 2007.
- [2] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, and F. Kanehiro, "Multi-contact vertical ladder climbing with an HRP-2 humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 561–580, 2016.
- [3] A. Paolillo, P. Gergondet, A. Cherubini, M. Vendittelli, and A. Kheddar, "Autonomous car driving by a humanoid robot," *Journal of Field Robotics*, 2017.
- [4] Y. Abe, M. da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, San Diego, California, 2-4 August 2007, pp. 249–258.
- [5] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts," in *IEEE/RAS International Conference on Humanoid Robots*, Pittsburgh, PA, November 29 - December 1 2007, pp. 81–88.
- [6] K. Bouyarmane and A. Kheddar, "Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, 25-30 September 2011.
- [7] S. Feng, X. Xinjilefu, W. Huang, and C. G. Atkeson, "3d walking based on online optimization," in *IEEE-RAS International Conference on Humanoid Robots*, 2013.
- [8] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014.
- [9] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.
- [10] P. E. E. Gill, S. J. Hammarling, W. Murray, M. A. Saunders, and M. H. Wright, "User's guide for lssol (version 1.0): a fortran package for constrained linear least-squares and convex quadratic programming," Stanford University, Standord, California 94305, Tech. Rep. 86-1, January 1986.
- [11] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, "Visual servoing in an optimization framework for the whole-body control of humanoid robots," *IEEE Robotics and Automation Letters*, 2017.
- [12] Z. Bien and J.-X. Xu, *Iterative Learning Control: Analysis, Design, Integration and Applications*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.
- [13] F. J. Harris, "On the use of harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51 – 83, January 1978.
- [14] L. Dong and G. Morel, "Robust trocar detection and localization during robot-assisted endoscopic surgery," in *IEEE International Conference on Robotics and Automation*, 2016.
- [15] S. Ahmad and C. N. Lee, "Shape recovery from robot contour-tracking with force feedback," *Advanced Robotics*, vol. 5, no. 3, pp. 257–273, 1990.
- [16] L. Mi and Y.-B. Jia, "High precision contour tracking with a joystick sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
- [17] M. Nitsche, T. Krajník, P. Čížek, M. Mejail, and T. Duckett, "WhyCon: an efficient, marker-based localization system," in *IEEE/RAS IROS Workshop on Open Source Aerial Robotics*, 2015.
- [18] A. Bolotnikova, K. Chappellet, A. Paolillo, A. Escande, G. Anbarjafari, A. Suarez-Roos, P. Rabaté, and A. Kheddar, "A circuit-breaker use-case operated by a humanoid in aircraft manufacturing," in *IEEE Conference on Automation Science and Engineering*, Xi'an, China, 20-23 August 2017.