UNIVERSITY OF PLYMOUTH

PEARL

Development of a dynamics model for the Baxter robot

Smith, Alex; Yang, Chenguang; Li, Chunxu; Ma, Hongbin; Zhao, Lijun

Link:
Link to publication in PEARL

Lagrange-Euler Closed Dynamics Model of the Baxter Robot

Alex Smith, Chenguang Yang, Xinyu Wang and Hongbin Ma

Abstract— The dynamic model of a robot is important to find the relation between the joint actuator torques and the resulting motion. There are two common methods to do this: The Lagrange formulation, which gives a closed form of the dynamic equations, and the Newton-Euler method, which uses a recursive form. Presented in this paper is a formulation of the Lagrange-Euler (L-E) equations representing the dynamics of the Baxter manipulator. These equations are then verified against torque trajectories recorded from the Baxter robot. Experimental studies show that torques generated using the L-E method closely match recorded actuator torques. All of Baxter's kinematic and dynamic parameters are presented here for easy future reference, and the full symbolic dynamics are made available online for closed loop analysis by the community.

I. INTRODUCTION

An accurate dynamic model of a robot manipulator is useful in many ways: for the design of motion control systems, analysis of mechanical design, simulation of manipulator motion, etc. Many control alorithms, such as computed torque control [1], predictive control [2] and sliding mode control [3] normally require an accurate model of the manipulator dynamics, commonly in the form:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau \qquad (1)$$

where q denotes the vector of joint angles, $M(q) \in \Re^{n \times n}$ is the symmetric, bounded, positive definite inertia matrix, and n is the degree of freedom (DoF) of the robot arm; $C(q,\dot{q})\dot{q} \in \Re^n$ denotes the Coriolis and Centrifugal force; $G(q) \in \Re^n$ is the gravitational force, and $\tau \in \Re^n$ is the vector of actuator torques. In this form the kinetic energy of the manipulator is described within $M(q)\ddot{q} + C(q,\dot{q})$, and the potential energy represented in the gravity term $G(q)$. This can then be used to calculate either the forward dynamics (useful for simulation), where the manipulator motion is calculated based on a vector of applied torques, or the inverse dynamics (useful for control design) where the torques for a given set of joint parameters can be calculated.

There are two commonly used methods for formulating the dynamics in eq. (1), based on the specific geometric and inertial parameters of the robot: the Lagrange-Euler (L-E) formulation and the Recursive Newton-Euler (RN-E) method

A. Smith, C. Yang and X. Wang are with the Centre for Robotics and Neural Systems, Plymouth University, Plymouth PL4 8AA, UK. X. Wang and H. Ma are with School of Automation, Beijing Institute of Technology, China. C. Yang is also with School of Automation Science and Engineering, South China University of Technology, China. Corresponding author is C. Yang. Email: cyang@ieee.org

TABLE I
NOMENCLATURE

n	Degrees of freedom (DoF) of the manipulator
$q, \dot{q}, \ddot{q} \in \Re^{n \times 1}$	Vector of joint position, angular velocity and acceleration respectively
a, d, α, θ	Variables denoting the Denavit-Hartenberg parameters
$I_i \in \Re^{3 \times 3}$	Inertia tensor of link i
m	Link mass
$\bar{r}_i \in \Re^{4 \times 1}$	Centre of mass of link i
$^iT_j \in \Re^{4 \times 4}$	Homogenous transform from link i to j

[4]. Both are equivalent, as they both describe the dynamic behaviour of the robot motion, but are specifically useful for different purposes.

The L-E method is based on simple, systematic methods to calculate the kinetic and potential energies of a rigid body system. The works of Bajeczy [5], [6], show that the equations of dynamic motion for a robot manipulator are highly non-linear, consisting of inertial and gravity terms, and are dependent on the link pysical parameters and configuration (i.e. position, angular velocity and acceleration). This provides the closed form of the robot dynamics, and is therefore applicable to the analytical computation of robot dynamics [7], and therefore can be used to design joint-space (or task-space, using transformation via the Jacobian) control strategies.

The L-E form may also be used for forward and inverse dynamic calculation, but this requires the calculation of a large number of coefficients in $M(q)$ and $C(q,\dot{q})$ from eq. (1), which may take a long time. This makes this method somewhat unsuitable for online dynamic calculations, especially as other methods such as RN-E (described in the next section), or Lee's Generalised d'Alembert Equations (GAE) [8] produce more simple, albeit messy, derivations which are much faster [4]. A recursive L-E method has also been described [9] which greatly reduces the computational cost of the L-E formulation and brings it into line with RN-E methods. The N-E formulation is based on a balance of all the forces acting on the generic link of the manipulator; this forms a set of equations with a recursive solution [10], and was developed . A forward recursion propagates link velocities and accelerations, then a backward recursion propagates the forces and torques along the manipulator chain. This was developed as a more efficient method than L-E, and is based on the principle of the manipulator being a serial chain; when a force is applied to one link, it may also produce motion in connected links. Due to this effect there may be considerable duplication of calculation [11], which can be avoided if expressed in a recursive form. This
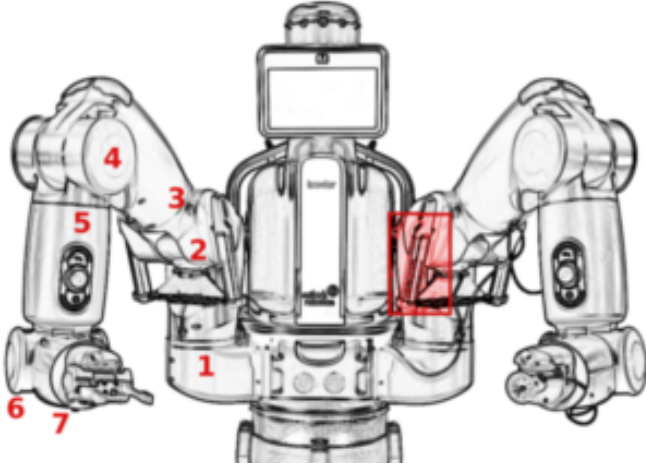
Fig. 1. Diagram of the Baxter robot. Joints *1, 2, 3* comprise the shoulder, *4 and 5* the elbow, and *6, 7* the wrist. The spring highlighted in *red*, two of which are attached to each arm, generate large forces to improve gravity compensation. Note, in this case, the arm on the left has a linear actuator at the end effector, while the arm on the left is equipped with a vacuum cup.

reduction in computational load greatly reduces calculation time, allowing the forward and inverse dynamics calculations to be performed in real time; therefore enabling real-time torque control methods of robot manipulators.

Our previous work related to the Baxter robot relied on the implementation from Peter Corke's Robotics Toolbox [12][13]. This, however, does not provide a closed loop solution of the dynamics, meaning that control analysis of the dynamic parameters is not possible. In addition, we found that it was not feasible to formulate symbolic representations of a higher-DoF robot such as the Baxter, due to the excessive memory consumption and computation time. In this work, we present the formulation and experimental verfication of the L-E equations for the Baxter robot, for analysis of the manipulator dynamics. A symbolic closed form of the dynamics has also been produced, which is available for public use and downloadable from `http://www.cgyang.net/downloads.html`. This was not previously available in any form online, and all kinematic and dynamic parameters are recorded in this paper in clear, standard forms, for easy future reference. It is hoped that this will prove useful for many researchers developing control architectures for the Baxter robot.

The common inertial and geometrical parameters required in derivation are described in table I.

II. THE BAXTER ROBOT

A recent creation from Rethink Robotics, the Baxter robot is comprised of two 7 Degree of Freedom (DoF) manipulators in a bimanual configuration, attached to a central pedestal (see fig. 1). The design aim of Baxter was to create a safe, flexible and affordable robot for integration into low-volume production [14][15]. The Software Development Kit (SDK) has also been released, opening Baxter up to research opportunities.

Link	θ	d (m)	a (m)	α (rad)	m (kg)
1	θ_1	0.2703	0.069	$-\pi/2$	5.70044
2	θ_2	0	0	$\pi/2$	3.22698
3	θ_3	0.3644	0.069	$-\pi/2$	4.31272
4	θ_4	0	0	$\pi/2$	2.07206
5	θ_5	0.3743	0.01	$-\pi/2$	2.24665
6	θ_6	0	0	$\pi/2$	1.60979
7	θ_7	0.2295	0	0	0.54218

One of the main features of Baxter is the Series Elastic Actuators (SEAs) which are present in every joint; these are comprised of a spring coupling between the motor and the link, with a built in hall effect to measure the deflection. This creates a naturally compliant design, but also means that the torque at each joint can be easily estimated by measuring the spring deflection and multiplying by the known stiffness constant.

There is also a gravity compensation controller running in a high frequency loop, which calculates the torques required to counteract gravitational pull on each joint as well as the forces applied to joint 2 from the large external springs highlighted in fig. 1. This is important to note, as the gravity compensation is applied by default, and must be manually disengaged for full torque control application.

A. Denavit-Hartenberg and Inertial Parameters of the Baxter Robot

The Denavit-Hartenberg (D-H) parameters and link masses of the Baxter manipulator are given in table II and are derived from the Universal Robot Descriptor File (URDF) [16]. These parameters describe the configuration of the links, and form the basis of the Lagrange-Euler formulation. The homogenous link transform matrices are formed from the D-H parameters as such:

$$
{}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{2}
$$

where ${}^0T_i = {}^0T_1\,{}^1T_2\ldots{}^{i-1}T_i$.

The inertia tensors of each joint are given in table III, represented by the inertias working in each axis I_{xx}, I_{yy}, I_{zz} and cross-talk inertia between axes I_{xy}, I_{yz}, I_{xz}. Here it is represented as a row vector, but is also commonly found in $I^{3\times3}$ symmetric matrix form.

The centre of mass (CoM) for each link is given in table IV, which forms the homogenous column vector $\bar{r}_i = [\bar{x}_i\ \bar{y}_i\ \bar{z}_i\ 1]^T$.

III. LAGRANGE-EULER FORMULATION

The Lagrange-Euler equations of motion for a conservative system [4] are given by

$$
L = K - P,
$$
$$
\tau = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}
\tag{3}
$$

Link	I_{xx}	I_{yy}	I_{zz}
1	0.0470910226	0.035959884	0.0376697645
2	0.027885975	0.020787492	0.0117520941
3	0.0266173355	0.012480083	0.0284435520
4	0.0131822787	0.009268520	0.0071158268
5	0.0166774282	0.003746311	0.0167545726
6	0.0070053791	0.005527552	0.0038760715
7	0.0008162135	0.0008735012	0.0005494148
Link	I_{xy}	I_{yz}	I_{xz}
1	-0.0061487003	-0.0007808689	0.0001278755
2	-0.0001882199	0.0020767576	-0.00030096397
3	-0.0039218988	-0.001083893	0.0002927063
4	-0.0001966341	0.000745949	0.0003603617
5	-0.0001865762	0.0006473235	0.0001840370
6	0.0001534806	-0.0002111503	-0.0004438478
7	0.000128440	0.0001057726	0.00018969891

Link	$\bar{x}$	$\bar{y}$	$\bar{z}$
1	-0.05117	0.07908	0.00086
2	0.00269	-0.00529	0.06845
3	-0.07176	0.08149	0.00132
4	0.00159	-0.01117	0.02618
5	-0.01168	0.13111	0.0046
6	0.00697	0.006	0.06048
7	0.005137	0.0009572	-0.06682

where K and P are the total kinetic and potential energies of the the system respectively, $q \in \Re^n$ is the generalised robot coordinates eqivalent to θ in table II, and τ is the generalised torque at the robot joints [4]. The kinematic and potential energies are given by:

$$K = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{i} \sum_{k=1}^{i} \left[\mathrm{Tr} \left(U_{ij} \, J_i \, U_{ik}^T \right) \dot{q}_j \dot{q}_k \right]$$

$$P = \sum_{i=1}^{n} -m_i \, \mathbf{g} \left({}^0T_i \, \bar{r}_i \right) \quad (4)$$

which, when substituted into eq. (3), gives the expression:

$$\tau_i = \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q}$$

$$= \sum_{j=i}^{n} \sum_{k=1}^{j} \mathrm{Tr}(U_{jk} J_j U_{ji}^T) \ddot{q}_k$$

$$+ \sum_{j=i}^{n} \sum_{k=1}^{j} \sum_{m=1}^{j} \mathrm{Tr}(U_{jkm} J_j U_{ji}^T) \dot{q}_k \dot{q}_m$$

$$- \sum_{j=i}^{n} m_j \mathbf{g} U_{ji} \bar{r}_j. \quad (5)$$

This can be expressed more simply in the form given in eq. (1), as a sum of the inertia, Coriolis/centrifugal and gravity terms. The elements of the symmetric matrix $M(q)$ are given by

$$M_{i,k} = \sum_{j=\max(i,k)}^{n} \mathrm{Tr}(U_{jk} \, J_j \, U_{ji}^T) \quad i,k = 1,2,\dots n, \quad (6)$$
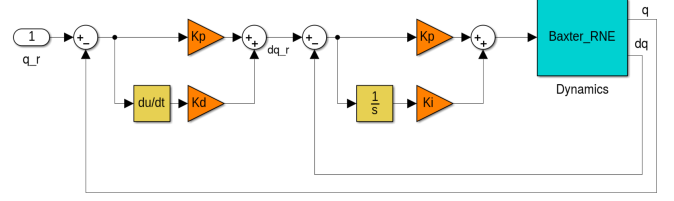

Fig. 2. Block diagram of torque control system.

the Coriolis/centrifugal force vector $C(q, \dot{q})$

$$C_i = \sum_{k=1}^{n} \sum_{m=1}^{n} h_{ikm} \dot{q}_k \dot{q}_m$$

$$h_{ikm} = \sum_{j=\max(i,k,m)}^{n} \mathrm{Tr}(U_{jkm} \, J_j \, U_{ji}^T) \quad (7)$$

and the gravity vector $G(q)$

$$G_i = \sum_{j=i}^{n} (-m_j \mathbf{g} U_{ij} \bar{r}_j) \quad (8)$$

where $\mathbf{g} = [0, \, 0, \, -9.81, \, 0]$ is the gravity row vector. The matrix U_{ij} is the rate of change of points on link i relative to the base as the joint position q_j changes

$$U_{ij} \equiv \frac{\partial T_i^0}{\partial q_j} = \begin{cases} {}^0T_{j-1} \, Q_j \, {}^{j-1}T_i & j \le i \\ 0 & j > i \end{cases} \quad (9)$$

which allows derivation of the interaction effects between joints, U_{ijk}

$$U_{ijk} \equiv \frac{\partial U_{ij}}{\partial q_k}$$

$$= \begin{cases} {}^0T_{j-1} \, Q_j \, {}^{j-1}T_{k-1} \, Q_k \, {}^{k-1}T_i & i \ge k \ge j \\ {}^0T_{k-1} \, Q_k \, {}^{k-1}T_{j-1} \, Q_j \, {}^{j-1}T_i & i \ge j \ge k \\ 0 & i < j \text{ or } i < k \end{cases} \quad (10)$$

where, for Baxter, as all joints are revolute,

$$Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (11)$$

The J_i matrices are independent of link position or motion, and therefore only needs to be calculated *once* from the inertia tensors, link masses and link CoMs:

$$J_i = \begin{bmatrix} \frac{-I_{xxi}+I_{yyi}+I_{zzi}}{2} & I_{xyi} & I_{xzi} & m_i\bar{x}_i \\ I_{xyi} & \frac{I_{xxi}-I_{yyi}+I_{zzi}}{2} & I_{yzi} & m_i\bar{y}_i \\ I_{xzi} & I_{yzi} & \frac{I_{xxi}+I_{yyi}-I_{zzi}}{2} & m_i\bar{z}_i \\ m_i\bar{x}_i & m_i\bar{y}_i & m_i\bar{z}_i & m_i \end{bmatrix} \quad (12)$$

This concludes the calculations required to form the L-E dynamics of the Baxter arm.

IV. EXPERIMENT

To collect data from the Baxter robot, a PID position controller is employed in a double loop configuration, as shown in fig. 2. A reference position q_r generates the outer loop:

$$e = q_r - q, \quad \dot{e} = \frac{\mathrm{d}}{dt} e$$
$$\dot{q}_r = K_p e + K_d \dot{e} \tag{13}$$

which is then used to generate the inner loop:

$$\dot{\epsilon} = \dot{q}_r - q_r, \quad \epsilon = \int \dot{\epsilon} \, \mathrm{d}t$$
$$\tau_r = K_p \dot{\epsilon} + K_i \epsilon. \tag{14}$$

The trajectories were created in two ways: generated using sine and cosine patterns or using a touchpad input, both in Cartesian space. Inverse kinematics are performed using the inverse Jacobian method, that is

$$\dot{q}_r = J^\dagger(q) \dot{x}_r \tag{15}$$

where $\dot{x}_r$ is the reference Cartesian velocity and $J^\dagger$ is the pseudo-inverse of the Jacobian matrix. The selected test trajectories, in fig. 3, show the actual Cartesian test trajectories x which are calculated from $x = F(q)$, where $F(q)$ is the forward kinematics of the robot. For the experiment, the right-hand manipulator of the Baxter was driven through these three trajectories and data collected at 50Hz, including joint positions and velocities $q, \dot{q}$, Cartesian position x and the estimated torques applied to the motors τ. This is calculated on board the Baxter from the deflection of the internal springs (and large external springs at joint 2), summed with the automatic gravity compensation. The joint accelerations $\ddot{q}$ were estimated through first order numerical differentiation of $\dot{q}$, which accounts for the noise in the calculated results.

V. RESULTS

To find the explicit form of Baxter manipulator dynamics, MATLAB's symbolic toolbox was utilised. In their raw state, the symbolic representations of the elements of $D(q), C(q, \dot{q})$ and $G(q)$ have many coefficients (over half a million), so cannot be printed here but are available online in MATLAB workspace format for analytical use.

To confirm the accuracy of the process, a numerical form was also created. Joint positions and velocities were recorded from the Baxter moving in three different trajectories, and the joint accelerations estimated through numerical differentiation, i.e. $\ddot{q}_i = \frac{\mathrm{d}\dot{q}_i}{dt}$, where $dt = 0.02$ is the sampling period of the trajectory recorder. The results from the L-E form are compared against torques recorded from the Baxter, and torque trajectories generated using the RN-E method from Peter Corke's Robotics Toolbox [12]. It is possible to generate the analytical representation of eq. (1) using this RN-E method, but only if n is small due to heavy memory consumption. Due to the way Baxter is controlled, the recorded torques are a sum of the actuator torques measured via internal spring deflection and two torque vectors acting on joint 2 (hysteresis and crosstalk) to compensate for
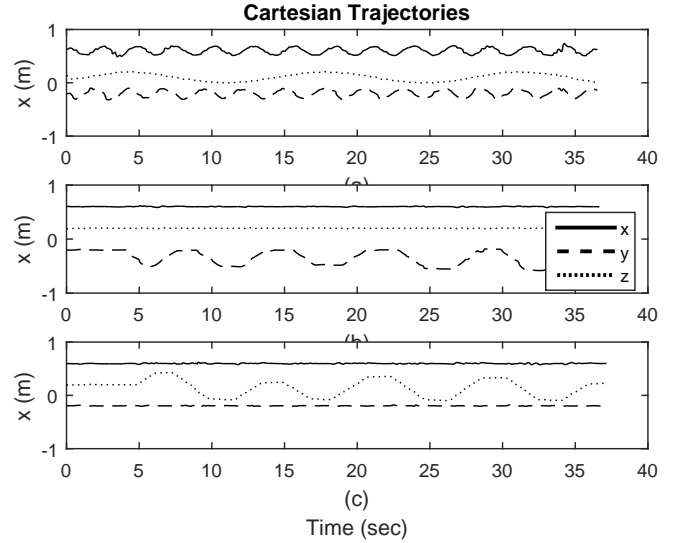

Fig. 3. Cartesian trajectories selected for experimentation. In (a) all three dimentions follow a sinusoidal pattern, with a lower frequency in the z-axis. For the trajectories in (b) and (c) the arm was moved only in the y and z-axes, respectively.

Fig. 4. Baxter robot in the test position.

the large external springs, mentioned previously. All results shown are collected from the right-hand manipulator, with the end effector aligned with the z-axis as in fig. 4. No external forces were applied to the arm during testing.

In fig. 5 the arm was moving in all three planes. It is noticable that the torques generated from L-E and RN-E are much noisier; this is due to the numerical differentiation of the joint accelerations $\ddot{q}$. This could be reduced by passing it through a low pass filter. However, we can see that the shape of the trajectories is very similar. The first joint and distal joints 5-7 only require very small torque input as they are mostly unaffected by gravity. Examining the L-E error plot in fig. 5, the noise dominates the largest errors but it is centred around zero for all joints, confirming that there are no bias errors. The RN-E error plot shows a similar range of error, but it is noticable that the error for joint 3 has some positive bias.

In fig. 6 we have similar results, with an even smaller error result. In this case the arm was moved in a way to generate
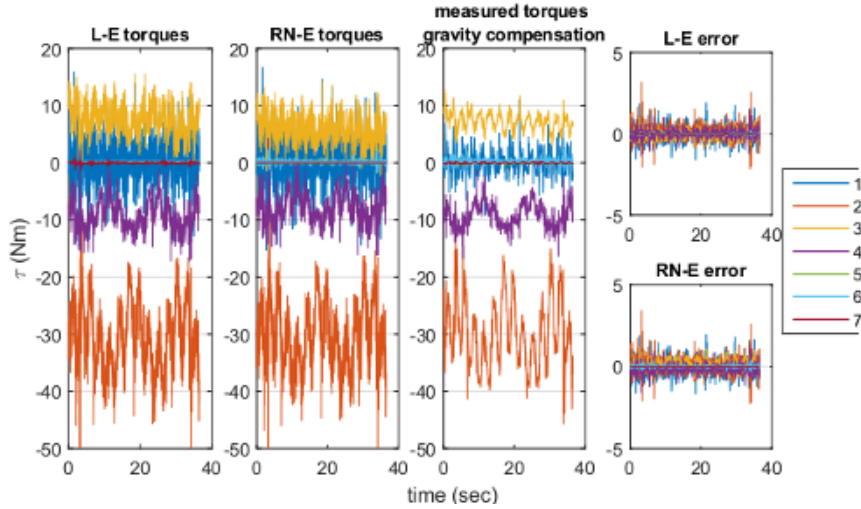
Fig. 5. Comparing torque generated through L-E and RN-E methods with torques recorded from the Baxter robot during the trajectory from fig. 3(a). The trajectory for this sample was moving the end-effector in a circular trajectory in the x, y planes and in a cosine pattern in the z-axis, where $x = 0.6 + 0.1\sin(t)$, $y = -0.2 + 0.1\cos(t)$ and $z = 0.1 + 0.1\cos(0.2t)$. The errors (*far right*) are the modeled torques subtracted from the recorded torques.
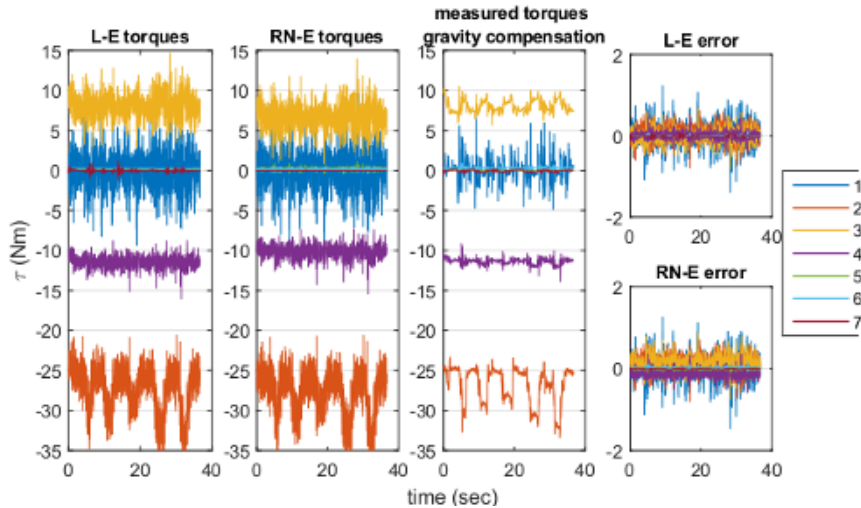

Fig. 6. Second comparison of torque trajectories from fig. 3(b); for this sample the end-effector is fixed in the x, z plane and is switched quickly between two positions in the y-axis.

higher accelerations by quickly switching the target position in the y-axis only. This movement is mostly achieved using joint 2 at the shoulder, noticable in the plots. The low error result in this case confirms a good match for the kinetic part of the dynamic model. Again, looking at the RN-E there is an obvious positive bias in the torques calculated for joint 3.

A slower trajectory was applied to the robot for the results in fig. 7, moving primarily in the z-axis, which is evident by the large changes occuring in joint 4. Noise is reduced due to minimal acceleration in the trajectory. The error results again show no bias errors, and within a good tolerance of around ± 1.5 Nm which mostly can be accounted for by noise from the acceleration trajectory derivation.

A good comparison of the methods is shown in table V, where the average (and a sum total) error of both methods

for each joint are shown side by side. These are calculated from each trajectory set, i.e. set 1 in table V corresponds to the first trajectory results in fig. 5. Looking through the table, it can be seen that the average error for each joint is comparable between the methods, apart from in joints 3-4 which have significantly larger errors in every set. This gives a good indication that the L-E method is not only accurate, but also slightly more accurate than the RN-E method.

VI. CONCLUSIONS

This paper describes the method for derivation of the Lagrange-Euler dynamics of the Baxter manipulator, followed by experimental verification using data collected from the robot.

Our results show that the derived model is a good match to the real dynamics, with low errors in three different end-
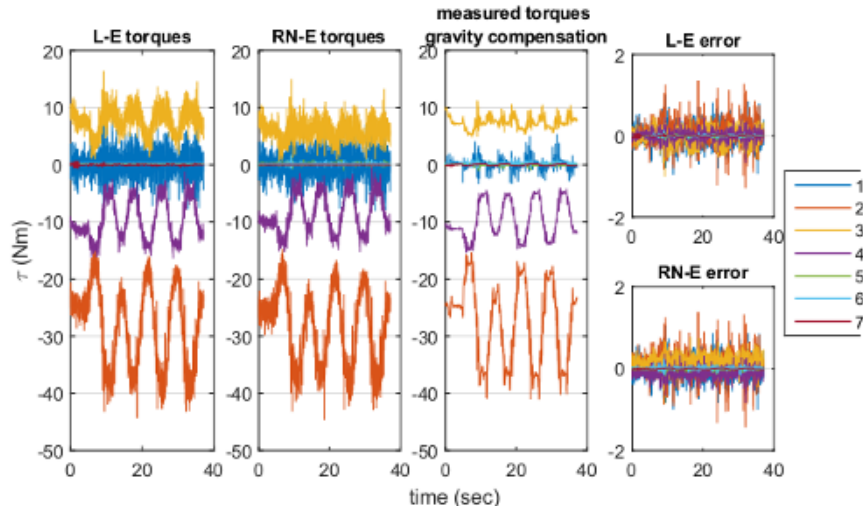
Fig. 7. Final comparison of torque trajectories from fig. 3(c) input. The end-effector was moved up and down in the z-axis, and held constant in the x, y plane.

TABLE V
AVERAGES OF CALCULATION ERRORS FOR L-E AND RN-E METHODS.

Joint	Set 1 L-E	Set 1 RN-E	Set 2 L-E	Set 2 RN-E	Set 3 L-E	Set 3 RN-E
$|\bar{e}_1|$	0.0105	0.0105	0.0148	0.0149	0.0061	0.0058
$|\bar{e}_2|$	0.1002	0.0665	0.0464	0.0931	0.0703	0.0872
$|\bar{e}_3|$	0.0475	0.1382	0.0083	0.1355	0.0367	0.1358
$|\bar{e}_4|$	0.0151	0.1231	0.0079	0.1210	0.0064	0.1148
$|\bar{e}_5|$	0.0068	0.0120	0.0099	0.0145	0.0079	0.0140
$|\bar{e}_6|$	0.0006	0.0103	0.0047	0.0136	0.0003	0.0113
$|\bar{e}_7|$	0.0012	0.0036	0.0003	0.0055	0.0013	0.0024
$\sum_{i=1}^{n} \bar{e}_i$	0.0710	0.0869	0.0527	0.1161	0.0366	0.1089

effector trajectories shown in figs. 5 to 7. The majority of the error can be attributed to high frequency noise generated from numerical derivation of joint accelerations. A smooth torque trajectory would be achieveable without this noise. The RN-E method from the Robotics Toolbox has a comparable performance when calculating torques numerically.

As mentioned earlier in section V, the analytical representation of the dynamics is very large before simplification, but available online in the form of a MATLAB workspace file. Currently, efforts are being made to reduce the number of coefficients without affecting the overall accuracy of the result. The authors hope that this document and the closed dynamic form will prove useful for researchers working with the Baxter robot, by compiling all relevant parameters in one place and removing the need for performing a lengthy calculation of the closed loop dynamics.

ACKNOWLEDGMENT

REFERENCES

[1] M. Uebel, I. Minis, and K. Cleary, "Improved computed torque control for industrial robots," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992, pp. 528–533.

[2] P. Poignet and M. Gautier, "Nonlinear model predictive control of a robot manipulator," in *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on*. IEEE, 2000, pp. 401–406.

[3] Y. Feng, X. Yu, and Z. Man, "Non-singular terminal sliding mode control of rigid manipulators," *Automatica*, vol. 38, no. 12, pp. 2159–2167, 2002.

[4] K. Fu and C. RC Gonzalez, "Robotics control, sensing, vision, and," 1987.

[5] A. K. Bejczy, "Robot arm dynamics and control," *Jet Propulsion Laboratory Technical Memo*, pp. 33–669, 1974.

[6] A. Bejczy and R. Paul, "Simplified robot arm dynamics for control," in *Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conference on*. IEEE, 1981, pp. 261–262.

[7] S. M. Megahed, *Principles of robot modelling and simulation*. John Wiley & Sons, Inc., 1993.

[8] C. Lee, B. Lee, and R. Nigam, "Development of the generalized d'alembert equations of motion for mechanical manipulators," in *Decision and Control, 1983. The 22nd IEEE Conference on*. IEEE, 1983, pp. 1205–1210.

[9] J. M. Hollerbach, "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 10, no. 11, pp. 730–736, 1980.

[10] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2009.

[11] P. Mckerrow, *Introduction to robotics*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[12] P. I. Corke, "A robotics toolbox for matlab," *Robotics & Automation Magazine, IEEE*, vol. 3, no. 1, pp. 24–32, 1996.

[13] A. M. Smith, C. Yang, H. Ma, P. Culverhouse, A. Cangelosi, and E. Burdet, "Novel hybrid adaptive controller for manipulation in complex perturbation environments," *PloS one*, vol. 10, no. 6, p. e0129281, 2015.

[14] "Rethink robotics," http://www.rethinkrobotics.com/, 2015, [Online; accessed 6 June 2015].

[15] C. Fitzgerald, "Developing baxter," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[16] Z. Ju, C. Yang, and H. Ma, "Kinematic modeling and experimental verification of Baxter robot," in *Proceedings of the 33rd Chinese Control Conference Nanjing, China, 28-30 Jul, 2014*. CCC, 2014, pp. 8518–8523.