# The FutureForge Manipulator and an Approximate Analytical Solution Algorithm for its Nonlinear Dynamics

Daniel Johnston*, Matthew Cartmell*, Bradley Wynne*, Vikram Pakrashi**, and Ivana Kovacic***

**Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, Scotland*

***UCD Centre for Mechanics, Dynamical Systems and Risk Laboratory, School of Mechanical and Materials Engineering, University College Dublin, Dublin, Ireland*

****CEVAS, Faculty of Technical Sciences, University of Novi Sad, Serbia*

**Abstract.** This paper examines a linearisation algorithm developed for the nonlinear dynamics of the *FutureForge* manipulator - part of a state-of-the-art forging platform being built in Glasgow, Scotland. Building on previous work, the authors investigate means of obtaining approximate analytical solutions for the highly nonlinear system with a view to developing a practical control system in future work. This is done by a combination of algebraic approximation methods, a practical understanding of the machine's operating characteristics, as well as the application of a perturbation method. The devised method is verified by comparing the resulting approximate analytical solutions with numerical solutions to the nonlinear problem.

**Keywords:** Linearisation, Industrial Manipulators

## 1 Introduction

This work follows on from a previous piece of work involving many of the same authors [1]. Previously, the authors examined the design of a large-scale industrial manipulator for the *FutureForge* project commissioned by the Advanced Forming Research Centre (AFRC) in Glasgow, Scotland. The manipulator's intended purpose is to handle metallic workpieces through their treatment in a 2,000-tonne hydraulic press: carrying them to and from the press, positioning them in the press appropriately, and ensuring their precise placement throughout these movements. Since this earlier work's publication, the manipulator has been successfully installed at the AFRC and a working control system has been implemented by Clansman Dynamics Ltd. The work we describe in this paper contributes to the development of a digital twin of the manipulator that will be used to train operators in a virtual reality environment.

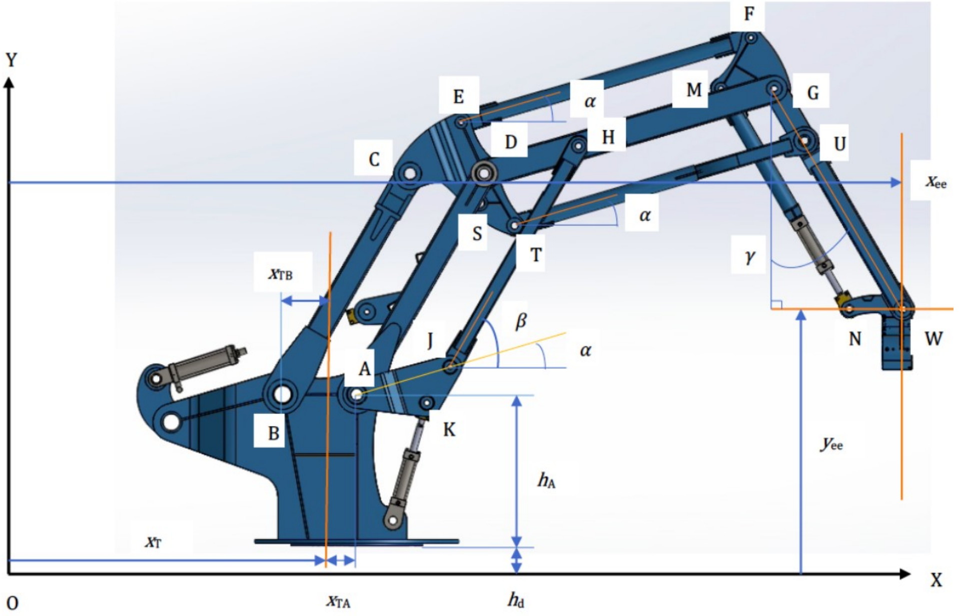

Figure 1: A schematic diagram of the *FutureForge* manipulator, depicting its configuration in its 2D planar workspace. This image previously appeared in the preceding work by Cartmell *et al.* [1].

Interestingly, the manipulator we are considering is rather unique in the field of industrial robotics. At first glance, it appears to be a serial system since the end-effector's position is governed by the positioning of three – apparently sequential – parallelogram linkages. Definitionally, serial manipulators are open kinematic chains of links, and, in many industrial examples, the motion of their end-effectors is actuated by revolute or prismatic actuators located at the joints between links

[2]. For configurations such as this, many powerful techniques have been developed to the end of solving the dynamics and kinematics involved. Such methods include Meldrum, Rodriguez, and Franklin's recursive approach to matrix inversion [3] and Saha's approach to decomposition of the inertia matrix [4]. Furthermore, several control approaches are based on this type of configuration also. These include Gazit and Widrow's "Back Propagation Through Links" method [5]; Ouyang and Zhang's virtual velocity-based method [6]; and, Talebpour and Namvar's application of adaptive control to serial manipulators in the context of satellite testing [7]. However, this manipulator may not be regarded as serial. Instead, the end-effector's position is driven by two actuators at its base. Therefore, the *FutureForge* manipulator might be thought of as a hybrid between a serial and a parallel manipulator. To be clear what we mean by this: the end-effector's position, in the planar workspace, is the result of a series of parallelogram configurations, these individually being closed-chain mechanisms; and, the configurations of each parallelogram are dictated by the parallelograms preceding them. We consider the linkage ordering as flowing from base to end-effector. In this way, the manipulator benefits from having a large workspace relative to its size and footprint [8]. Additionally, the parallelogram structure of the linkages allows it to carry heavier workpieces with greater accuracy [9]. The trade-off in realising these benefits is that the resulting system is significantly more complex to model analytically.

## 1.1 Summary and Influence of Previous Work

In the work preceding this [1], the authors discussed how the manipulator was intended to operate. Due to its novel design, actuating the hydraulics connected to link AD moves the end-effector purely horizontally. Similarly, the hydraulic actuator connected to point K causes vertical motion in the end-effector. One significant point to note, however, is that this "vertical actuation" does involve some variance in the end-effector's horizontal position such that its path is somewhat curved due to physical constraints of the mechanism. The authors derived the governing dynamics of the manipulator via Lagrangian mechanics and were able to solve this model numerically.

The model that resulted is expressed as a system of two governing equations: one for actuation of the cylinder connected at K (the "vertical case"); and, one for actuation of the cylinder connected to AD (the "horizontal case"). The generalised coordinates in the equations are the angles controlled by these two actuating cylinders: $\alpha(t)$, for the vertical actuation angle; and, $\beta(t)$, for the horizontal actuation angle. Thus, later in this present work when we refer to the "vertical actuation problem", we simply mean the ODE which is derived from the Lagrange equation

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \alpha'(t)}\right) - \frac{\partial L}{\partial \alpha} = Q_v \tag{1}$$

where $Q_v$ is the generalised force applied through this cylinder at K. Similarly, the "horizontal actuation problem" refers to the ODE resulting from

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \beta'(t)}\right) - \frac{\partial L}{\partial \beta} = Q_h \tag{2}$$

in which $Q_h$ is the generalised force applied through the cylinder connected to AD. Note that, in the above equations and throughout this paper, we use the prime notation to denote differentiation with respect to time, staying consistent with the previous work mentioned, thus $\alpha' = \frac{d}{dt}\alpha(t)$, and $\beta' = \frac{d}{dt}\beta(t)$.

The ODEs that result from these two applications of Lagrange's equation are highly nonlinear and exhibit coupling between the two actuation coordinates $\alpha(t)$ and $\beta(t)$. Moreover, they are incredibly cumbersome to manage analytically to the point of being intractable - the curious reader is referred to Eqs (44) and (79) in [1]. As such, the present work was begun to attempt a linearisation strategy which could yield approximate analytical solutions to the underlying dynamics. The resulting linearised model will then be used in a control system for the manipulator – to be implemented in a virtual reality (VR) model of the manipulator. Since the manipulator's operation will always be controlled by a human operator, it is the role of this mathematical model and control system to replicate effectively the physical behaviour of the manipulator as accurately as possible where the physical machine must be capable of submillimetre accuracy in end-effector placement.

At the end of the previous work, we outlined a means of simplifying these aforementioned equations. Firstly, the two governing equations are simplified by collecting their terms into the following groups (and arranging them in this order in Eqs (3) and (4)): linear and nonlinear inertia, linear and quadratic damping, nonlinear restoring force, and excitation. Applying these groupings, we simplify the equations considerably. For the vertical actuation problem, the resulting form of the governing equation is

$$a_1\alpha'' + a_2\cos(2\alpha)\alpha'' + a_3\alpha' + a_4\sin(2\alpha)(\alpha')^2 + a_5\cos(\alpha) + a_6\sin(\alpha) = Q_v, \tag{3}$$

and the horizontal actuation problem is

$$b_1\beta'' + b_2\cos(2\beta)\beta'' + b_3\beta' + b_4\sin(2\beta)(\beta')^2 + b_5\cos(\alpha) + b_6\sin(\beta) = Q_h, \tag{4}$$

where the design specifications of the manipulator contribute to the constant $a$-terms in Eq (3) and $b$-terms in Eq (4). These specifications detail the individual component masses, lengths, and mass moments of inertia. From this point, the previous work introduced a small perturbation parameter, $\varepsilon$, into the governing dynamics via these constant terms. A development of this small parameter's introduction is elaborated on in Section 2.1. Then, we continue through Sections 2.2-2.4 to describe the linearisation process we have adopted. Finally, we discuss our results and conclusions in Sections 3 and 4 respectively.

# 2 Linearisation of the Governing Equations

## 2.1 Introduction of the Small Parameter

In the original governing equations, Eqs (3) and (4), the constant $a$ and $b$-terms are functions of the design specifications of the manipulator: component lengths, masses, mass moments of inertia, and geometric offsets. By computing these specifications to find the numerical values of all $a$ and $b$-terms, we can compare their magnitudes to introduce a small parameter, $\varepsilon$. This parameter is introduced to constants deemed significantly smaller than others. By this, we mean terms which are at least 100-times smaller than the largest constant in the same equation. Based on this approach, for Eq (3) we find that

$$a_2 = \varepsilon\bar{a}_2, \quad a_3 = \xi_{jf} + \varepsilon\xi_{af}, \quad a_4 = \varepsilon\bar{a}_4, \quad \text{and} \quad a_6 = \varepsilon\bar{a}_6. \tag{5}$$

Additionally, for Eq (4) we find

$$b_2 = \varepsilon\bar{b}_2, \quad b_3 = \varepsilon\xi_{jf} + \varepsilon^2\xi_{af}, \quad b_4 = \varepsilon\bar{b}_4, \quad \text{and} \quad b_6 = \varepsilon\bar{b}_6. \tag{6}$$

In the previous work by Cartmell *et al.*, the linear damping was manually introduced into the governing equations to account for both the viscous joint friction and aerodynamic drag in a single "lumped"-parameter [1]. Now notice that we have also expanded the definition of these linear damping coefficients $a_3$ and $b_3$. These expanded definitions split the two sources of drag so that they may be treated independently and can contribute to the $O(\varepsilon^0)$ and $O(\varepsilon^1)$ perturbation equations more appropriately. Notice the key difference between the substitutions in (5) and (6) is that the linear damping coefficient contributes to a higher order of perturbation equation for the horizontal problem than in the vertical problem. This is consistent with our strategy for introducing $\varepsilon$, however, it may be desirable to electively allow linear damping to contribute to the generating, $O(\varepsilon^0)$, problem for the horizontal actuation problem.

## 2.2 Transcendental-Algebraic Conversion

Referred to as the "TAC", the transcendental-algebraic conversion step is intended to pave the way for the substitution of asymptotic expansions of $\alpha$ and $\beta$. In this step, each transcendental term is replaced with an algebraic approximation derived via appropriate Taylor Series expansions. For the vertical actuation problem, we compute Taylor Series expansions of $\cos(2\alpha)$, $\sin(2\alpha)$, $\cos(\alpha)$, and $\sin(\alpha)$. The centres of these expansions are at $\alpha = 0$ radians, the centre of the operational range of $\alpha$, to benefit from the expansions' accuracy in both the positive and negative $\alpha$ directions. A MATLAB algorithm has been constructed to determine the minimum order of approximation required to ensure a maximum error of 0.1% in these approximations versus their respective target functions. Based on discussions with our industrial collaborators (the AFRC), this error requirement may be tightened or relaxed. Regardless, the method progresses in an identical fashion. The same approach is implemented for the horizontal problem, this time with the expansions centred at $\beta = \frac{\pi}{2}$ radians, the centre of the operational range of $\beta$.

To study the errors introduced by this step, we compute the numerical solutions of Eqs (3) and (4) and compare them to the numerical solutions of the approximated equations given by the TAC step. The impact of this approximation can be seen in each of the cases examined in Section 3 and is referred to in the included graphs as "Num (post-TAC)". Given $\alpha \in [-0.305, 0.305]$ radians and $\beta \in [\frac{\pi}{3}, \frac{2\pi}{3}]$ radians, the errors depicted in Figures 2-17 are very small indeed. Nonetheless, given how these errors grow with time, the orders of approximations may require further considerations if very long durations of manoeuvres are to be simulated

## 2.3 Exploitation of the Excitation Function

Until this point, the excitations applied to the vertical and horizontal cases were defined as constants, $Q_v$ and $Q_h$ respectively. Here we define them as nonlinear functions of both the relevant actuation angle and time. This approach is not dissimilar to Hsu's exploitation of strategically tuned excitation functions, according to the brief description in Hsu's work in [10]).

For the manipulator to resist falling under its own weight, part of the total excitation must go towards balancing the gravitational restoring force. This component of the excitation is denoted by $R_v(\alpha)$ and $R_h(\beta)$ for the vertical and horizontal problems respectively. However, for there then to be motion, there must also be an excitation beyond this gravity-balancing component. We represent this with a time dependent component in the two cases as $F_v(t)$ and $F_h(t)$ respectively. Thus, the revised definitions of the excitations are

$$Q_v = R_v(\alpha) + F_v(t) \tag{7}$$

for the vertical problem, and

$$Q_h = R_h(\beta) + F_h(t) \tag{8}$$

for the horizontal problem. Note that, to carry out its intended role, $R_v(\alpha)$ must equal the vertical restoring force.

$$R_v(\alpha) = a_5 \cos(\alpha) + a_6 \sin(\alpha) \tag{9}$$

The same is similarly true for the horizontal equivalent, $R_h(\beta)$, and the restoring force in the horizontal problem. Therefore, these two components of the excitations will balance any occurrence of the restoring forces throughout the perturbation equations, not just in the generating problem. This is also true for the horizontal case. Additionally, the time-dependent excitations, $F_v(t)$ and $F_h(t)$, will only contribute to the generating solutions. Note that other authors have previously adopted methods that involve assuming the absence of a gravitational restoring force [11]. In this section, we have provided a justification for this from an operational standpoint, rather than simply stating it as an *a priori* assumption.

## 2.4 Extraction of the Perturbation Equations from the Simplified Model

With the governing equations now suitably prepared, we can introduce one final pair of substitutions, the asymptotic approximations of $\alpha(t)$ and $\beta(t)$

$$\alpha(t;\varepsilon) \approx \alpha_0(t) + \varepsilon\alpha_1(t) \tag{10}$$
$$\beta(t;\varepsilon) \approx \beta_0(t) + \varepsilon\beta_1(t). \tag{11}$$

Substituting these into the revised governing equations and collecting the $O(\varepsilon^0)$ and $O(\varepsilon^1)$ terms, we find the following equations:

$$a_{0,2,1}\alpha_0'' + a_{0,1,1}\alpha_0' = a_{0,0,1}F_v \tag{12}$$

$$\begin{aligned} a_{1,2,1}\alpha_1'' + a_{1,1,1}\alpha_1' =& \big(a_{1,0,1} + a_{1,0,2}\alpha_0^2\big)\alpha_0'' \\ & - a_{1,0,3}\alpha_0' + \big(a_{1,0,4}\alpha_0 + a_{1,0,5}\alpha_0^3\big)\big(\alpha_0'\big)^2 \end{aligned} \tag{13}$$

$$b_{0,2,1}\beta_0'' = b_{0,0,1}F_h \tag{14}$$

$$\begin{aligned} b_{1,2,1}\beta_1'' =& \big(b_{1,0,1} + b_{1,0,2}\beta_0 + b_{1,0,3}\beta_0^2\big)\beta_0'' \\ & + b_{1,0,4}\beta_0' + \big(b_{1,0,5} + b_{1,0,6}\beta_0 + b_{1,0,7}\beta_0^2 + b_{1,0,8}\beta_0^3\big)\big(\beta_0'\big)^2. \end{aligned} \tag{15}$$

Eqs (12) and (14) are the generating problems for the vertical and horizontal cases respectively. Similarly, Eqs (13) and (15) are the $O(\varepsilon^1)$ dynamics of the vertical and horizontal cases. Note the subscripts utilised for the $a$ and $b$ constants in Eqs (12) to (15). These are used to usefully identify components of the solution and to identify their origin in the problem. This method of encoding the constants' origins and occurrences is inspired by [12] but is distinct from this approach as our work is significantly more limited although serving our purpose adequately. The three numbers used have different functions: the first refers to the $\varepsilon$-order of the problem that the constant first appears in; the second refers to the order of derivative multiplying the constant; and, the third is a simple counter. For example, the constant $a_{0,2,1}$ (i.e., with the subscripts 0, 2, and 1) is found in the $O(\varepsilon^0)$ problem, is the coefficient of the 2nd derivative of the dependent variable, and is the 1st coefficient with these previous two designators. By contrast, the constant $b_{1,0,5}$ is the 5th constant to occur in the $O(\varepsilon^1)$ horizontal problem on the right-hand side of the equation. Note this feature of the "0"-designator in our notation: since we have removed all occurrences of the gravitational restoring force from all levels of the manipulator's dynamics, the "0"-designator is not reserved for the coefficients of the dependent variables and can be freely used for terms arranged on the right-hand sides of Eqs (12)-(15).

As was mentioned briefly previously, a key difference between the horizontal and vertical problems is the relative insignificance of linear damping in the former. The result of this is seen in comparing the two generating problems. For the sake of illustration, we continue this paper with the assumption that linear damping should not contribute to the horizontal problem at the $O(\varepsilon^0)$ level. If this assumption needed to be changed, then the solution process for the horizontal case precisely follows the algorithm of the vertical case. From this point, the solution of the linearised system is relatively simple to compute analytically. As intended, both $O(\varepsilon^0)$ problems may be solved independently of their $O(\varepsilon^1)$ counterparts.

# 3 Results

To inspect the quality of our proposed linearisation algorithm, we compare the approximate analytical (AA) solutions to the numerical solutions of the governing equations at three points in the process. These three points for finding numerical solutions are: before any linearisation takes place, Eqs (3) and (4); immediately after the TAC step is undertaken; and, once the governing problems are stated as perturbation equations. We undertake four cases to examine, these are detailed in Table 1. Cases 1 and 2 consider relatively slow motions of the end-effector; while, Cases 3 and 4 consider faster, more abrupt motions. At the time of writing, we are seeking data from our industrial collaborators that describes manoeuvres undertaken in physical testing of the machine. When this becomes available, we will compare the model against this. However, for now we are only concerned with the efficacy of the linearisation process.

| Case | $\varepsilon$ | Initial Conditions | | | | Vertical Actuation | | Horizontal Actuation | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\alpha(0)$ (rad) | $\alpha'(0)$ (rad/s) | $\beta(0)$ (rad) | $\beta'(0)$ (rad/s) | $F_v$ (kN) | $t_v$ (s) | $F_h$ (kN) | $t_h$ (s) |
| 1 | 0.001 | -0.3054 | 0 | 1.0472 | 0 | 3 | 3 | 5 | 3 |
| 2 | 0.001 | 0.3054 | 0 | 2.0944 | 0 | -3 | 3 | -5 | 3 |
| 3 | 0.001 | -0.3054 | 0 | 1.0472 | 0 | 30 | 1 | 50 | 1 |
| 4 | 0.001 | 0.3054 | 0 | 2.094 | 0 | -30 | 1 | -50 | 1 |

Table 1: Cases examined to test the suitability of the linearisation algorithm in the absence of physical test data.

Using the kinematic equations of the earlier work [1], we can inspect the impact of the errors found on the accuracy of the end-effector's placement. Using the expressions for the end-effector's Cartesian position, we can see that changes of over 1.8×$10^{-4}$ radians in $\beta$ result in the end-effector motion exceeding 1mm. Given our previously stated intent that the control system we will go on to implement should be accurate on a sub-millimetre scale, we can acknowledge that the absolute error in $\beta$ should be less than this value. For $\alpha$, the equivalent condition on the error is found as ±3×$10^{-4}$ radians. This is also why the initial positions in Table 1 are rounded to 4 decimal places (since 0.0001 is less than either the $\alpha$ or $\beta$ tolerance stated).

## 3.1 Case 1

In Case 1, we see the following approximations of the vertical actuation problem arise from our linearisation process. Note that the numerical values included in the equations listed in the following equation, and the rest of Section 3, are rounded to 5 significant figures since this is consistent with the initial conditions.

$$\begin{aligned}\alpha(t) \approx &- 88.573 - 1.1043 \times 10^{-8}e^{-0.1699t} + 6.0005 \times 10^{-8}e^{-0.13595t} - 1.1744 \times 10^{-7}e^{-0.10196t}\\ &+ 3.5945 \times 10^{-7}e^{-0.067974t} + 88.2694e^{-0.033987t} - 0.0017626e^{-0.033987t} + 3.0000t\\ &- 1.8765 \times 10^{-9}te^{-0.13595t} + 8.7830 \times 10^{-9}te^{-0.10196t} - 5.7442 \times 10^{-9}te^{-0.067974t}\\ &- 3.0007 \times 10^{-5}te^{-0.033987t} - 1.9186 \times 10^{-9}t^2 - 1.2755 \times 10^{-10}t^2e^{-0.10196t}\\ &+ 5.7531 \times 10^{-10}t^2e^{-0.067974t} - 7.6792 \times 10^{-10}t^2e^{-0.033987t}\\ &+ 1.7371 \times 10^{-11}t^3 - 4.3351 \times 10^{-12}t^3e^{-0.067974t} + 8.7297 \times 10^{-12}t^3e^{-0.033987t}\\ &- 7.3669 \times 10^{-14}t^4 - 1.4734 \times 10^{-13}t^4e^{-0.033987t}\end{aligned} \tag{16}$$

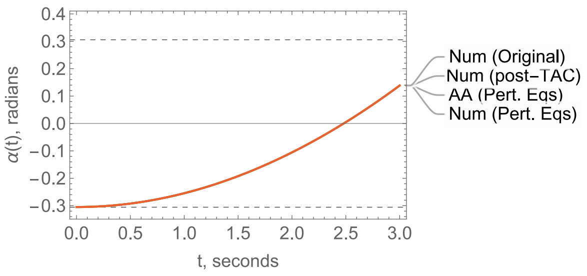

Figure 2: Approximate solutions of the vertical problem for Case 1. Note the operational limits of $\alpha$ marked at ±0.3054.

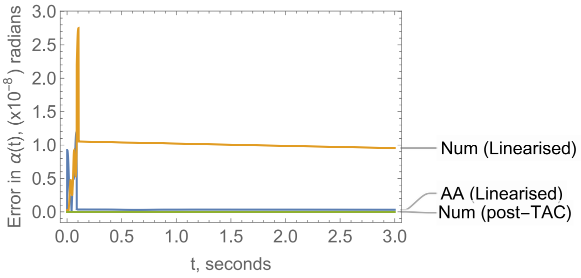

Figure 3: The absolute error for each approximation in Fig 2 versus "Num (Original)" as a reference.

As can be seen, with a cursory glance at Fig 2, as well as in Fig 3, the approximate analytical solution and two numerical solutions are extremely close to the numerical solution of the original problem. The reason for the numerical solution of the linearised problem, "Num (Linearised)", having a slightly larger error versus the target solution is simply that the numerical method used naturally introduces numerical errors where the analytical method does not.

For the horizontal actuation problem, we initially see another very close resemblance between the four illustrated solutions in Fig 4. Note that the approximate analytical solution for this is given by

$$\begin{aligned}\beta(t) \approx& 1.0472 + 0.040710t^2 - 2.5948 \times 10^{-4}t^3 - 2.1092 \times 10^{-4}t^4 \\ &+ 3.7226 \times 10^{-6}t^6 + 9.6805 \times 10^{-8}t^8 - 1.6916 \times 10^{-9}t^{10}.\end{aligned} \tag{17}$$

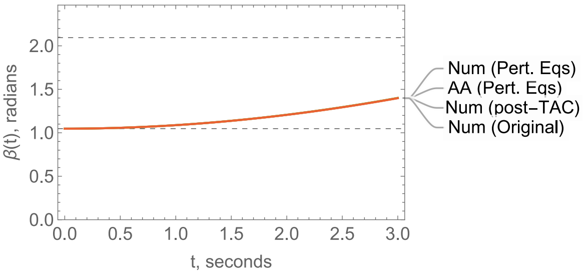

Figure 4: Approximate solutions of the horizontal problem for Case 1. Note the operational limits of $\beta$ at $\frac{\pi}{3}$ and $\frac{2\pi}{3}$.

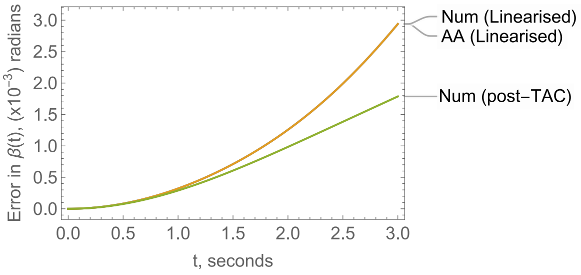

Figure 5: The absolute error for each approximation in Fig 4 versus "Num (Original)" as a reference.

However, under closer inspection, we see that the absolute error in each of these is significantly higher than in the vertical problem. There are two factors that contribute to this reduced accuracy through the linearisation process. The first, most significant, of these is the order of Taylor Series approximation used in the TAC step. This will have a significant impact on the horizontal actuation problem since the operational range of $\beta$ is significantly larger. The second of these reasons is the lack of linear damping included in the generating problem. Although these two factors clearly contribute to reduced accuracy of the model, only a comparison with physical testing data will rule whether they are acceptable trade-offs or not. If they are found to be unacceptable, we can simply increase the approximation order at the TAC stage or be sure to include linear damping in the generating dynamics.

### 3.2 Case 2

Similarly to Case 1, we see greater accuracy in the vertical actuation problem than in the horizontal problem although the latter may still be satisfactorily accurate. The vertical approximate analytical solution for this is simply the solution for Case 1 multiplied by $(-1)$. Whereas, that of the horizontal actuation problem is

$$\begin{aligned}\beta(t) \approx& 2.09440 - 0.040710t^2 + 2.5949 \times 10^{-4}t^3 + 2.1092 \times 10^{-4}t^4 \\ &- 3.7226 \times 10^{-6}t^6 - 9.6805 \times 10^{-8}t^8 + 1.6916 \times 10^{-9}t^{10}\end{aligned} \tag{18}$$

where we note that $\beta(0)$ is simply the new initial condition and all other terms are the negatives of those in Case 1.

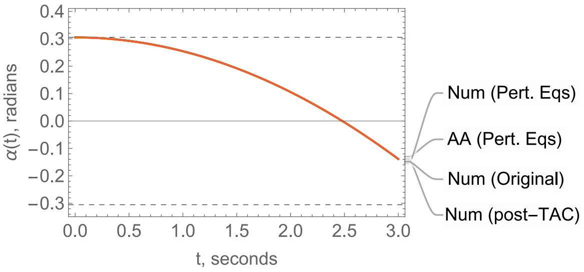

Figure 6: Approximate solutions of the vertical problem for Case 2. Note the operational limits of $\alpha$ marked at ±0.305.

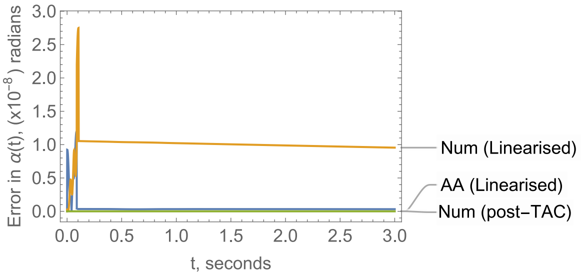

Figure 7: The absolute error for each approximation in Fig 6 versus "Num (Original)" as a reference.

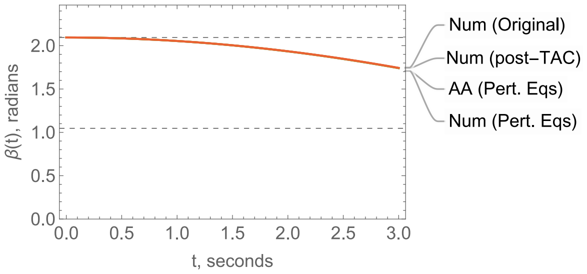

Figure 8: Approximate solutions of the horizontal problem for Case 2. Note the operational limits of $\beta$ at $\frac{\pi}{3}$ and $\frac{2\pi}{3}$.

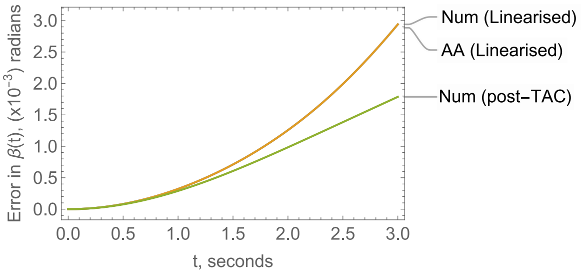

Figure 9: The absolute error for each approximation in Fig 8 versus “Num (Original)” as a reference.

## 3.3 Case 3

Case 3, as previously stated, is intended to be a more aggressive version of Case 1. What we are interested in here is how the approximations behave with a reduced duration of manoeuvre and with higher driving forces. The approximate analytical solution for the vertical actuation problem is as follows.

$$
\begin{aligned}
\alpha(t) \approx & - 883.29 - 0.0011043e^{-0.16993t} + 0.0059833e^{-0.13595t} - 0.011665e^{-0.10196t} \\
& + 0.035894e^{-0.067974t} + 882.69e^{-0.033987t} + 0.26223e^{-0.033987t} \\
& + 30.012t - 1.8765 \times 10^{-4}te^{-0.13595t} + 8.7596 \times 10^{-4}te^{-0.10196t} - 5.6412 \times 10^{-4}te^{-0.067974t} \\
& - 0.0010491te^{-0.033987t} \\
& - 1.9138 \times 10^{-4}t^2 - 1.2755 \times 10^{-5}t^2e^{-0.10196t} + 5.7412 \times 10^{-5}t^2e^{-0.067974t} \\
& - 7.6558 \times 10^{-5}t^2e^{-0.033987t} \\
& + 1.7344 \times 10^{-6}t^3 - 4.3351 \times 10^{-7}t^3e^{-0.067974t} + 8.6763 \times 10^{-7}t^3e^{-0.033987t} \\
& - 7.3669 \times 10^{-9}t^4 - 1.4734 \times 10^{-8}t^4e^{-0.033987t}
\end{aligned} \tag{19}
$$

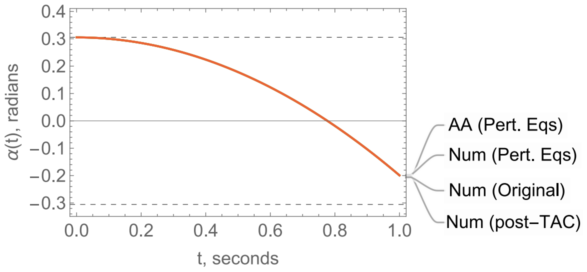

Figure 10: Approximate solutions of the vertical problem for Case 3. Note the operational limits of $\alpha$ marked at $\pm 0.305$.

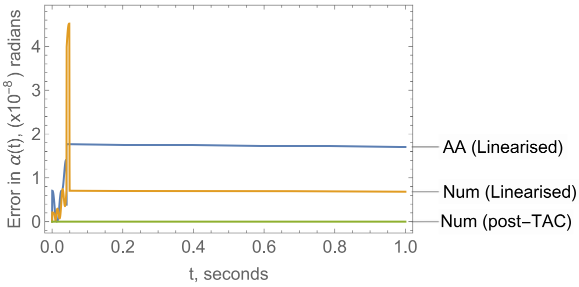

Figure 11: The absolute error for each approximation in Fig 10 versus “Num (Original)”.

We see, for the vertical problem, that the resulting errors are all extremely small as in previous results. Note that the only difference in this case is that the numerical solution of the fully linearised problem is now lower than the approximate analytical solution. The errors depicted here are so low, however, that we are not concerned with this subtle change.

Interestingly, the horizontal results are very similar to those of Case 1 too. This reinforces the idea that these errors can be reduced by a higher-order approximation in the TAC step – since the accuracy of this step is dependent on the physical range, in $\beta$, of the manoeuvre rather than on its duration directly. The analytical approximation for this is found to be as follows.

$$
\begin{aligned}
\beta(t) \approx & 1.0472 + 0.40710t^2 - 0.0025948t^3 - 0.021092t^4 \\
& + 0.0037226t^6 + 9.6805 \times 10^{-4}t^8 - 1.6916 \times 10^{-4}t^{10}
\end{aligned} \tag{20}
$$

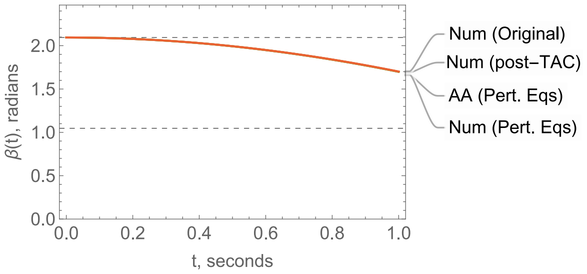

Figure 12: Approximate solutions of the horizontal problem for Case 3. Note the operational limits of $\beta$ at $\frac{\pi}{3}$ and $\frac{2\pi}{3}$.

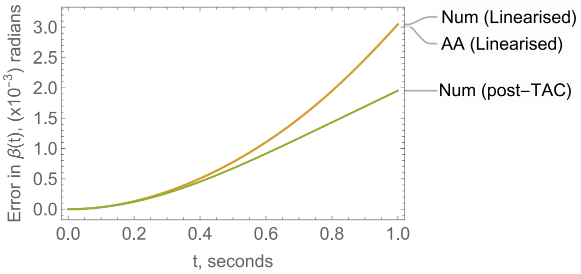

Figure 13: The absolute error for each approximation in Fig 12 versus "Num (Original)".

## 3.4 Case 4

For Case 4, both vertical and horizontal actuation problems give very similar results to Case 3. The approximate analytical solution for the vertical actuation problem for Case 4 is equal to that of Case 3 multiplied by $(-1)$; whereas, the horizontal problem is

$$\begin{aligned}\beta(t) \approx & 2.0944 - 0.40710t^2 + 0.0025948t^3 + 0.021092t^4 \\ & - 0.0037226t^6 - 9.6805 \times 10^{-4}t^8 + 1.6916 \times 10^{-4}t^{10}\end{aligned} \tag{21}$$

where the constant term is the value of the initial $\beta$-condition for Case 4 and the rest of the approximation is identical to that of Case 3 multiplied by $(-1)$.

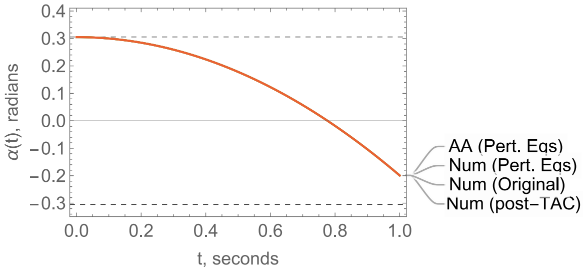

Figure 14: Approximate solutions of the vertical problem for Case 4. Note the operational limits of $\alpha$ at $\pm 0.305$.

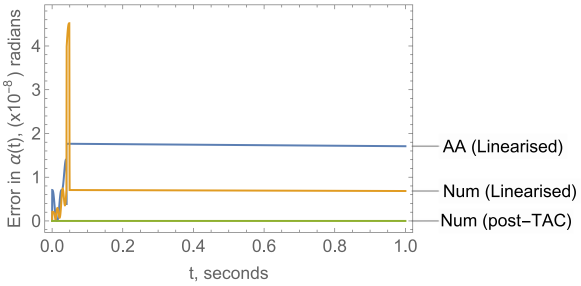

Figure 15: The absolute error for each approximation in Fig 14 versus "Num (Original)".

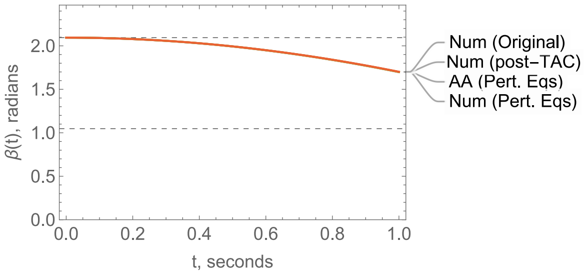

Figure 16: Approximate solutions of the horizontal problem for Case 4. Note the operational limits of $\beta$ at $\frac{\pi}{3}$ and $\frac{2\pi}{3}$.

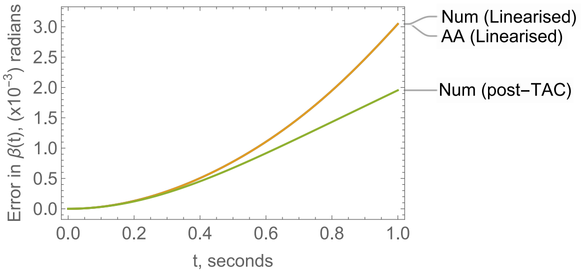

Figure 17: The absolute error for each approximation in Fig 16 versus "Num (Original)".

From the graphs illustrating the absolute errors in the horizontal problems for Cases 1 and 2, all errors are less than the $\beta$ tolerance value ($1.8 \times 10^{-4}$ radians) up to 0.7 seconds. And, for Cases 3 and 4, all errors are less than this value up to 0.25 seconds. Through all of the examined cases, the error in $\alpha$ is below $5 \times 10^{-5}$ radians throughout the time domains - well within the previously stated tolerance of $3 \times 10^{-4}$ radians. These observations can be used to show that that the linearisation algorithm supports the requirement of submillimetre accuracy in end-effector placement – particularly given the sharp, abrupt motions involved in manoeuvring a workpiece through the active forge.

# 4 Conclusion and Comments for Continuing Work

In this work, we have presented an algorithm for the linearisation of the governing dynamics of a large-scale industrial manipulator. The manipulator - as part of the *FutureForge* project commissioned by the AFRC - has a unique design that rules-out many established techniques which are designed around serial or parallel manipulators. Previous work led to the derivation of the nonlinear governing equations we have considered, and continuing collaboration with our industrial partners will see the presented algorithm undergo further fine-tuning. The presented algorithm sees a small perturbation parameter, $\varepsilon$, introduced to the governing dynamics via the relative magnitude of the constant coefficients. Then, the transcendental terms are approximated via algebraic expressions using appropriately ordered Taylor Series expansions. Observations about the machine's practical operations are then made to remove the restoring forces from consideration at all levels of the resulting perturbation problems. We then inspected the results achieved by comparing our approximate analytical solutions against numerical solutions to the original, nonlinear governing dynamics. By comparison with these numerical solutions, we also illustrated the errors introduced at the TAC stage and at the conclusion of the linearisation process. We found that the approximate analytical solutions for the vertical actuation problem are well-within the acceptable range of accuracy for the context of our work. The errors found in the horizontal actuation problem are notably larger, but these may still be acceptable pending comparison with physical test data from our industrial collaborators. In the instance that they are above the acceptable limits of error, we have noted suitable courses of action in Section 3.

Going forward, we are actively seeking to compare our simulations against the physical test data gathered by our industrial collaborators. In parallel with this, we are in the process of designing a suitable, robust control system for implementation in the digital twin model of this manipulator.

***Acknowledgements.*** We wish to acknowledge the work of Clansman Dynamics Ltd (based in East Kilbride, Scotland) in constructing the *FutureForge* manipulator on behalf of the AFRC and for providing the necessary design specifications so that we could carry out our own work. We also acknowledge the funding contribution provided by Clansman Dynamics Ltd to the PhD studentship of D. Johnston. Additionally, V. Pakrashi would like to acknowledge Science Foundation Ireland Centre I-FORM.

# References

[1] M. Cartmell, I. Gordon, D. Johnston, S. Liang, L. McIntosh, and B. Wynne, *Modelling the dynamics of a large-scale industrial manipulator for precision control*, Advanced Problems in Mechanics, Russian Academy of Sciences, 2021 [in print].

[2] Y. Chen, G. Leitmann, and J. Chen, "Robust control for rigid serial manipulators: A general setting," in *American Control Conference*, vol. 2, IEEE, 1998, 912–916 vol.2, ISBN: 0743-1619. DOI: 10.1109/ACC.1998.703540.

[3] D. Meldrum, G. Rodriguez, and G. Franklin, "An order (n) recursive inversion of the jacobian for an n-link serial manipulator," in *IEEE International Conference on Robotics and Automation*, vol. 2, IEEE Comput. Soc. Press, 1991, pp. 1175–1180. DOI: 10.1109/ROBOT.1991.131768.

[4] S. Saha, "A decomposition of the manipulator inertia matrix," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 301–304, 1997, ISSN: 1042-296X. DOI: 10.1109/70.563652.

[5] R. Gazit and B. Widrow, "Backpropagation through links: A new approach to kinematic control of serial manipulators," in *Tenth International Symposium on Intelligent Control*, IEEE, 1995, pp. 99–104, ISBN: 2158-9860. DOI: 10.1109/ISIC.1995.525044.

[6] F. Ouyang and T. Zhang, "Virtual velocity vector-based offline collision-free path planning of industrial robotic manipulator," *International journal of advanced robotic systems*, vol. 12, no. 9, p. 129, 2015, ISSN: 1729-8806 1729-8814. DOI: 10.5772/60127.

[7] M. Talebpour and M. Namvar, "Zero-gravity emulation of satellites in present of uncalibrated sensors and model uncertainties," in *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, IEEE, 2009, pp. 1063–1068, ISBN: 1085-1992. DOI: 10.1109/CCA.2009.5280698.

[8] S. Ajwad, J. Iqbal, M. Ullah, and A. Mehmood, "A systematic review of current and emergent manipulator control approaches," *Frontiers of Mechanical Engineering*, vol. 10, no. 2, pp. 198–210, 2015, ISSN: 2095-0241. DOI: 10.1007/s11465-015-0335-0.

[9] Y. Patel and P. George, "Parallel manipulators applications - a survey," *Modern Mechanical Engineering*, vol. 2, pp. 57–64, 2012. DOI: 10.4236/mme.2012.23008.

[10] I. Kovacic, *Nonlinear Oscillations*, 1st ed. Springer, Cham, 2020, pp. X, 273. DOI: 10.1007/978-3-030-53172-0.

[11] K. Lee and G. Chirikjian, “A new perspective on o(n) mass-matrix inversion for serial revolute manipulators,” in *2005 IEEE International Conference on Robotics and Automation*, vol. 2005, IEEE, 2005, pp. 4722–4726, ISBN: 1050-4729. DOI: 10.1109/ROBOT.2005.1570849.

[12] N. Motazedi, M. Cartmell, and J. Rongong, “Extending the functionality of a symbolic computational dynamic solver by using a novel term-tracking method,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 232, no. 19, pp. 3439–3452, 2017, ISSN: 0954-4062. DOI: 10.1177/0954406217737104.