

# Muscle decomposition and recruitment criteria influence muscle force estimates

L. Joakim Holmberg and Anders Klarbring

**Linköping University Post Print**

N.B.: When citing this work, cite the original article.

The original publication is available at [www.springerlink.com](http://www.springerlink.com):

L. Joakim Holmberg and Anders Klarbring, Muscle decomposition and recruitment criteria influence muscle force estimates, 2012, Multibody system dynamics, (28), 3, 283-289.

<http://dx.doi.org/10.1007/s11044-011-9277-4>

Copyright: Springer Verlag (Germany)

<http://www.springerlink.com/?MUD=MP>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-71021>

# Muscle decomposition and recruitment criteria influence muscle force estimates

L. Joakim Holmberg · Anders Klarbring

Received: 2 March 2011 / Accepted: 16 September 2011

**Abstract** It has recently been pointed out that muscle decomposition influence muscle force estimates in musculoskeletal simulations. We show analytically and with numerical simulations that this influence depends on the recruitment criteria. Moreover, we also show that the proper choices of force normalization factors may

---

This study was sponsored in part by the Swedish National Centre for Research in Sports (Grant No. 168/09).

DOI 10.1007/s11044-011-9277-4

L.J. Holmberg

Division of Mechanics, Institute of Technology, Linköping University, SE-581 83 Linköping, Sweden

E-mail: joakim.holmberg@liu.se

L.J. Holmberg

Swedish Winter Sports Research Centre, Mid Sweden University, SE-831 25 Östersund, Sweden

A. Klarbring

Division of Mechanics, Institute of Technology, Linköping University, SE-581 83 Linköping, Sweden

---

overcome the issue. Such factors for the *minmax* and the *polynomial* criteria are presented.

**Keywords** force normalization factor · *minmax* optimization criteria · musculoskeletal simulation · *polynomial* optimization criteria

## 1 Introduction

Recently, Blajer et al. [1] published an interesting article concerning the influence of selected modeling and computational issues on muscle force estimates. This topic is important for users of biomechanical simulations because it can serve as an aid in the development of "best practice". Using a planar arm model (comprising 2 segments, 2 joints and 8 muscles) Blajer et al. [1] estimated the muscle forces by inverse dynamics and static optimization. They compared results due to differences in coordinate systems, muscle paths, muscle decomposition and muscle recruitment optimization criteria. What we found intriguing was the influence of muscle decomposition on force estimates. When decomposing the *biceps brachii* muscle into two muscles with equal strength (half of the original) having identical origin and insertion points they found that the load sharing between muscles had changed. The force estimates changed for all muscles that played the same role as *biceps brachii* (arm flexors in this case). This behavior has not been noted in our own simulation work. But we also note that Blajer et al. [1] used a *polynomial* criteria while we normally use a *minmax* criteria, both described in [2].

The issue of muscle decomposition is important in several respects. For instance, it is common to model muscles as line objects, but many muscles (e.g. the deltoids) have wide origin or insertion points (or both). The normal solution

is then to decompose the muscle into several pieces. It seems important to know whether such modeling practice has unexpected and perhaps unwanted effects.

The aim is to study whether the influence of muscle decomposition on force estimates depends on the muscle recruitment optimization criteria.

## 2 Theory

In this section we study a simple small size muscle recruitment model based on static optimization. Only two muscles are originally involved and one of these are decomposed into two parts. By comparing in this way a two-muscle model to a three-muscle one, we are able to deduce how force normalization factors should be chosen in order to have a correct correlation between the two models. After an initial discussion of the general case of an arbitrary number of forces, the *minmax* criteria and the *polynomial* criteria are studied separately for the small size problem.

### 2.1 Optimization criteria

Two mathematical forms of the cost function that has been used in static optimization muscle recruitment models are, the *minmax* function

$$G(\mathbf{f}^m) = \max \left[ \frac{f_1^m}{N_1}, \dots, \frac{f_i^m}{N_i}, \dots, \frac{f_n^m}{N_n} \right] \quad (1)$$

and the *polynomial* one

$$G(\mathbf{f}^m) = \sum_{i=1}^n \left( \frac{f_i^m}{N_i} \right)^p, \quad (2)$$

where  $\mathbf{f}^m$  is the muscle force vector,  $f_i^m$  is individual muscle force,  $N_i$  is a normalizing factor representing the available strength (maximal force) of each individual

muscle,  $n$  is the number of muscles and  $p$  is the power of the polynomial. One of these functions are to be minimized under constraints of (dynamic) force equilibrium. It may be noted that as  $p$  goes to infinity the *polynomial* function should approach the *minmax* one.

## 2.2 Small size problems

*minmax* – we initially study a simplified problem involving two forces  $f_1$  and  $f_2$  and a single equilibrium equation. That is,

$$\begin{cases} \min_{f_1, f_2} \max \left[ \frac{f_1}{N_1}, \frac{f_2}{N_2} \right] \\ \text{subject to } f_1 + f_2 = r, \quad f_1 \geq 0, \quad f_2 \geq 0, \end{cases} \quad (3)$$

where the equation of the constraint is the equilibrium equation and  $r$  consists of inertial and external forces which are considered known at this stage. There is no lack of generality in not using arbitrary coefficients in front of the forces in this equation. However, an interpretation of such a simplified case could be that the two muscles have the same moment arm. The solution of problem (3) is easily shown to be (assuming  $r > 0$ )

$$f_1 = \frac{rN_1}{N_1 + N_2}, \quad f_2 = \frac{rN_2}{N_1 + N_2}.$$

Next, the second force  $f_2$  is replaced by a pair of forces  $f_2^1$  and  $f_2^2$  with available strengths  $N_2^1$  and  $N_2^2$ , respectively, resulting in the following modified problem:

$$\begin{cases} \min_{f_1, f_2^1, f_2^2} \max \left[ \frac{f_1}{N_1}, \frac{f_2^1}{N_2^1}, \frac{f_2^2}{N_2^2} \right] \\ \text{subject to } f_1 + f_2^1 + f_2^2 = r, \quad f_1 \geq 0, \quad f_2^1 \geq 0, \quad f_2^2 \geq 0. \end{cases} \quad (4)$$

The solution of this modified problem becomes

$$f_1 = \frac{rN_1}{N_1 + N_2^1 + N_2^2}, \quad f_2^1 = \frac{rN_2^1}{N_1 + N_2^1 + N_2^2}, \quad f_2^2 = \frac{rN_2^2}{N_1 + N_2^1 + N_2^2},$$

so

$$f_2^1 + f_2^2 = \frac{r(N_2^1 + N_2^2)}{N_1 + N_2^1 + N_2^2},$$

and we conclude that as long as  $N_2 = N_2^1 + N_2^2$ , it holds that  $f_2 = f_2^1 + f_2^2$ , which is what we would demand from an appropriated muscle decomposition.

*polynomial* – as for the *minmax* objective we formulate and compare the results of two problems:

$$\left\{ \begin{array}{l} \min_{f_1, f_2} \left[ \left( \frac{f_1}{N_1} \right)^p + \left( \frac{f_2}{N_2} \right)^p \right] \\ \text{subject to } f_1 + f_2 = r, \quad f_1 \geq 0, \quad f_2 \geq 0 \end{array} \right. \quad (5)$$

and

$$\left\{ \begin{array}{l} \min_{f_1, f_2^1, f_2^2} \left[ \left( \frac{f_1}{N_1} \right)^p + \left( \frac{f_2^1}{N_2^1} \right)^p + \left( \frac{f_2^2}{N_2^2} \right)^p \right] \\ \text{subject to } f_1 + f_2^1 + f_2^2 = r, \quad f_1 \geq 0, \quad f_2^1 \geq 0, \quad f_2^2 \geq 0. \end{array} \right. \quad (6)$$

For simplicity it is assumed that  $N_2^1 = N_2^2 \equiv \bar{N}$ . This implies that problem (6) becomes symmetric in the two second forces, which can therefore be assumed to be equal. We set  $2f_2^1 = 2f_2^2 \equiv \bar{f}$  and rewrite (6) as follows:

$$\left\{ \begin{array}{l} \min_{f_1, \bar{f}} \left[ \left( \frac{f_1}{N_1} \right)^p + 2^{(1-p)} \left( \frac{\bar{f}}{\bar{N}} \right)^p \right] \\ \text{subject to } f_1 + \bar{f} = r, \quad f_1 \geq 0, \quad \bar{f} \geq 0. \end{array} \right. \quad (7)$$

Thus, if

$$N_2^1 = N_2^2 = 2^{\frac{1-p}{p}} N_2 \quad (8)$$

problems (5) and (6) are identical and, therefore,  $f_2 = f_2^1 + f_2^2$ .

In conclusion, for the *minmax* objective any decomposition of the force normalization factor that sum to the original value gives a behavior that is what one would expect from the physical interpretation of the problem. For the *polynomial* objective, on the other hand, the value of the new normalization factors depends on the degree of the polynomial, and if one makes the natural choice of taking values that sum to the original value, one cannot expect to obtain forces that sum to the original force. Even though these conclusions are here derived for a simple small size problem, numerical results indicate that they are generally valid.

### 3 Numerical verification

To verify the force normalization factor (8) in Sec. 2.2, a similar model (Fig. 1) as the one used in [1] was created in the AnyBody modeling system 4.1 (AnyBody Technology A/S, Aalborg, Denmark). This software uses non-conventional musculoskeletal inverse dynamics with static optimization, in which the muscle forces are solved directly from body motion and external forces [3]. The approach uses a full set of Cartesian co-ordinates for each body segment in the system and the Newton-Euler equations of motion. Thus, the method used here is not exactly the same as in [1], but it should behave similarly.

Muscle data can be seen in Table 1 and the complete code can be seen in Online Resource 1. The two segments are rigid body elements, joints are ideal hinges and muscles have constant strength (i.e. there is no contraction dynamics). The simulated movement was arm flexion, generated by driving the shoulder and elbow joints with constant velocity. Start and end positions can be seen in Fig. 1.

Table 1: Muscle data

Muscle	Strength (N)
<i>m1:brachioradialis</i>	112.5
<i>m2:pronator teres</i>	225
<i>m3:brachialis</i>	375
<i>m4:biceps brachii</i>	562.5
<i>m41:biceps brachii - part 1</i>	$N^{m4}/2$ or $2^{(1-p)/p} N^{m4}$
<i>m42:biceps brachii - part 2</i>	$N^{m4}/2$ or $2^{(1-p)/p} N^{m4}$
<i>m5:triceps brachii - caput longum</i>	375
<i>m6:triceps brachii - caput mediale+laterale</i>	675
<i>m7:deltoid extensor part</i>	1125
<i>m8:deltoid flexor part</i>	1125

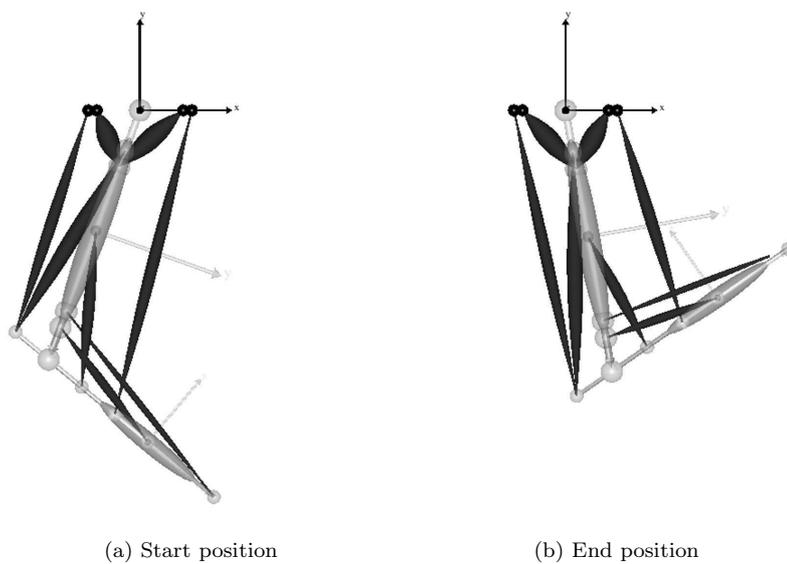


Fig. 1: The simulation model

Two set-ups of the model were created. The original, with one *biceps brachii* ( $m4$ ); and the modified, with *biceps brachii* decomposed into two muscles ( $m41$  and  $m42$ ) having the same origin and insertion points. To verify (8) for the simplified problem in Sec. 2.2, we first locked the shoulder joint and removed all muscles except  $m3$  and  $m4$  (or  $m41$  and  $m42$  for the modified set-up). We then used the full model to verify (8) for a more general case. Simulations were carried out using cost functions according to (1) and (2), the latter for several values of  $p$ .

#### 4 Results and Discussion

As seen in Fig. 2, 3 and 4 the *minmax* solution do not yield any differences, but the *polynomial* one may. Force estimates of all including flexors (as well as the deltoid extensor ( $m7$ )) and joint reactions change between set-ups when using the *polynomial* criteria if not choosing  $N$  according to (8). Note that (8) seems to be generally valid as numerical results from the full model (Fig. 3) are practically identical between the original set-up and the modified set-up when (8) is used. As expected, when  $p$  grows, the *polynomial* solutions resemble the *minmax* solutions. Interestingly, when  $p \rightarrow \infty$  in (8), that relation converges to  $N/2$ , which is the correct choice for *minmax* when decomposing one muscle into two muscle pieces of equal strength. The discontinuity in Fig. 2 for  $p = 1$  comes from a change in the numbers of muscles being active. As time increases there is a change from one to two muscles with non-zero muscle force. Note that when only one muscle is active the minmax objective is no different from the polynomial objective, which is seen in the fact that all curves coincide for the initial time interval. There is a similar situation in Fig. 3 for  $p = 1$ , although more complicated.

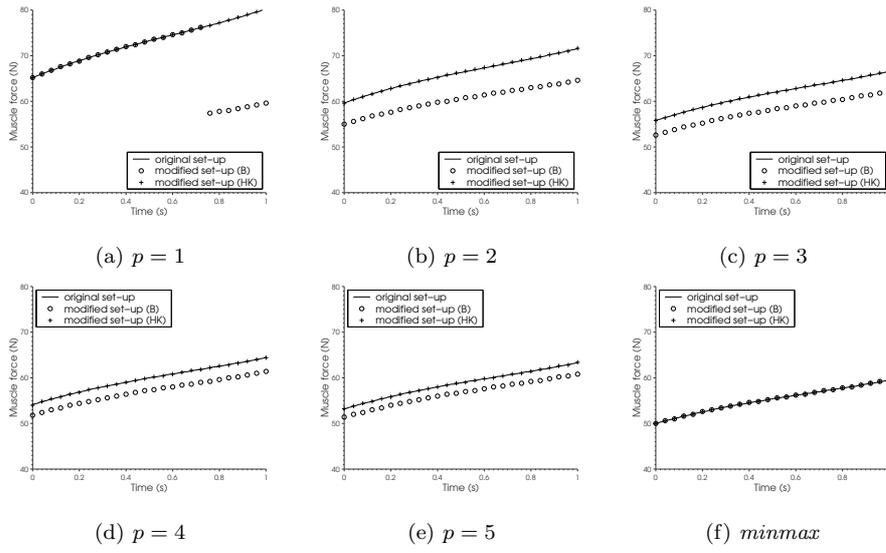


Fig. 2: Muscle force for  $f_2$  (original set-up) and  $f_2^1 + f_2^2$  (modified setup (B) with  $N_2^1 = N_2^2 = N_2/2$  according to [1] or modified set-up (HK) with  $N_2^1 = N_2^2 = 2^{(1-p)/p}N_2$  according to (8) in Sec. 2.2) at different values of  $p$  and  $minmax$  for the simplified problem in Sec. 2.2

There are several ways to construct a cost function in musculoskeletal modeling. Muscle force based cost functions are common, but not the only possibility, see e.g. [4]. In our study, the cost functions are based on muscle activation, i.e. normalized muscle force. These may be called "fatigue-like" criteria [5]. The value of the force normalization factor  $N$  is based on muscle physiological cross-section area and it seems logical to divide the strength of a muscle equal to the number of pieces, or at least that the new normalization factors would sum to the original value. But as shown, this only works for the  $minmax$  criteria and not the  $polynomial$  criteria. However, if the cost function includes muscle volume scaling in addition to muscle force normalization, the criteria can be characterized

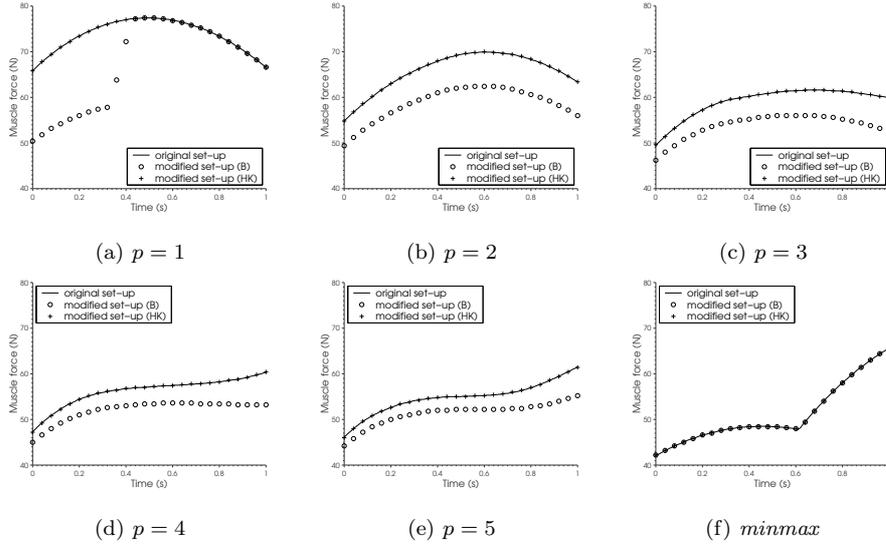


Fig. 3: Muscle force for *biceps brachii*,  $m_4$  (original set-up) and  $m_{41}+m_{42}$  (modified setup (B) with  $N^{m41} = N^{m42} = N^{m4}/2$  according to [1] or modified set-up (HK) with  $N^{m41} = N^{m42} = 2^{(1-p)/p} N^{m4}$  according to (8) in Sec. 2.2), at different values of  $p$  and  $minmax$  for the full model

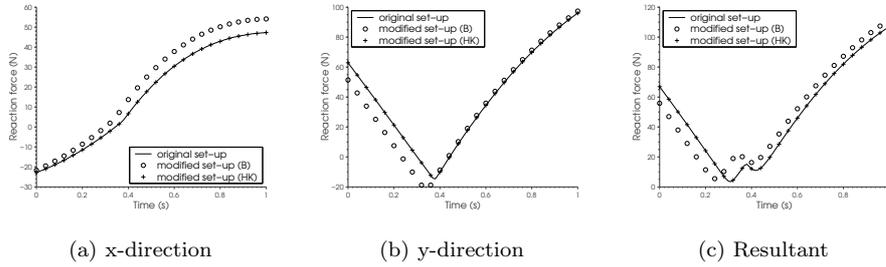


Fig. 4: Shoulder joint reactions at  $p = 2$  for original set-up, modified set-up (B) with  $N^{m41} = N^{m42} = N^{m4}/2$  according to [1] and modified set-up (HK) with  $N^{m41} = N^{m42} = 2^{(1-p)/p} N^{m4}$  according to (8) in Sec. 2.2 (x- and y-direction according to Fig. 1)

as "effort-like" [5]. According to a review [4], "fatigue-like" criteria are the most commonly used within inverse dynamics and static optimization. A notable exception is [6]. In the case of "effort-like" criteria, consistent muscle decomposition would be achieved for  $p = 1$ , not for  $p \rightarrow \infty$  (*minmax*). The reason for this is that  $N$  would then be proportional to the muscle volume divided by the muscle cross-sectional area and a natural decomposition would be  $N_2^1 = N_2^2 = N_2$ , i.e.  $p = 1$  in (8). Nevertheless, when Ackermann and van den Bogert [5] compared optimality principles for gait modeling, a cost function corresponding to the *minmax* criteria performed better than a cost function corresponding to a polynomial criteria, regardless of whether volume scaling was included or not (that model did not comprise any muscle decompositions).

To sum up, this study shows that force estimates may be influenced by muscle decomposition depending on the recruitment criteria. To overcome this, muscle decomposition force normalization factors for a *minmax* and a *polynomial* criteria are presented in Sec. 2.2. As Blajer et al. [1] show, there may be several issues to consider in biomechanical modeling. Having one less issue to worry about may add confidence in simulation results.

**Acknowledgements** The authors wish to acknowledge the insightful comments made by the reviewers.

## References

1. Blajer, W., Czaplicki, A., Dziewiecki, K., Mazur, Z.: Influence of selected modeling and computational issues on muscle force estimates. *Multibody Syst. Dyn.* **24**(4), 473–492 (2010). DOI 10.1007/s11044-010-9216-9.

2. Rasmussen, J., Damsgaard, M., Voigt, M.: Muscle recruitment by the min/max criterion—a comparative numerical study. *J. of Biomech.* **34**(3), 409–415 (2001). DOI 10.1016/S0021-9290(00)00191-3
3. Damsgaard, M., Rasmussen, J., Christensen, S.T., Surma, E., de Zee, M.: Analysis of musculoskeletal systems in the AnyBody Modeling System. *Simul. Model. Pract. and Theory* **14**(8), 1100–1111 (2006). DOI 10.1016/j.simpat.2006.09.001
4. Erdemir, A., McLean, S., Herzog, W., van den Bogert, A.J.: Model-based estimation of muscle forces exerted during movements. *Clin. Biomech. (Bristol, Avon)*. **22**(2), 131–154 (2007). DOI 10.1016/j.clinbiomech.2006.09.005
5. Ackermann, M., van den Bogert, A.J.: Optimality principles for model-based prediction of human gait. *J. Biomech.* **43**(6), 1055–1060 (2010). DOI 10.1016/j.jbiomech.2009.12.012
6. Happee, R., van der Helm, F.C.T.: The control of shoulder muscles during goal directed movements, an inverse dynamic analysis. *J. Biomech.* **28**(10), 1179–1191 (1995). DOI 10.1016/0021-9290(94)00181-3

```

// ESM_1.pdf
// AnyScript code, for use with the AnyBody modeling system 4.1

// Model used for the paper "Muscle decomposition and recruitment criteria
// influence muscle force estimates" in the journal Multibody System Dynamics
// DOI 10.1007/s11044-011-9277-4

// L. Joakim Holmberg & Anders Klarbring, Division of Mechanics,
// Institute of Technology, Linkoping University, Sweden
// Correspondence: joakim.holmberg@liu.se

// Muscle "m4" is biceps brachii while "m41" & "m42" are the new "pair" of
// biceps brachii with half the strength of the original (or based on p)

```

```

Main = {

    // The actual body model goes in this folder
    AnyFolder ArmModel = {

        // Global Reference Frame
        AnyFixedRefFrame GlobalRef = {

            AnyDrawRefFrame DrwGlobalRef = {
                ScaleXYZ = {0.1, 0.1, 0.1};
                RGB = {0,0,0};
            };
            AnyRefNode S = {
                sRel = {0,0,0};
            };
            AnyRefNode m8 = {
                sRel = {0.05,0,0};
                AnyDrawNode drw = {
                    Visible = On;
                    ScaleXYZ = {0.007, 0.007, 0.007};
                    RGB = {0, 0, 0};
                };
            };
            AnyRefNode m7 = {
                sRel = {-0.05,0,0};
                AnyDrawNode drw = {
                    Visible = On;
                    ScaleXYZ = {0.007, 0.007, 0.007};
                    RGB = {0, 0, 0};
                };
            };
            AnyRefNode m4 = {
                sRel = {0.06,0,0};
                AnyDrawNode drw = {
                    Visible = On;
                    ScaleXYZ = {0.007, 0.007, 0.007};
                    RGB = {0, 0, 0};
                };
            };
            AnyRefNode m5 = {
                sRel = {-0.06,0,0};
                AnyDrawNode drw = {
                    Visible = On;

```

```

        ScaleXYZ = {0.007, 0.007, 0.007};
        RGB = {0, 0, 0};
    };
};
}; // Global reference frame

```

```
// Segments
```

```

AnyFolder Segs = {
    AnySeg S1 = {
        r0 = {0, 0.3, 0};
        Axes0 = RotMat(-90*pi/180, z);
        Mass = 3.2;
        Jii = {0.001, 0.035, 0.035};
        AnyDrawSeg drw = {
            Opacity = 0.4;
            InertiaScale = 0.5;
        };
    };

```

```

    AnyRefNode S = {
        sRel = {-0.15, 0, 0};
    };

```

```

    AnyRefNode E = {
        sRel = {0.16, 0, 0};
    };

```

```

    AnyRefNode m2 = {
        sRel = {0.12, 0, 0};
    };

```

```

    AnyRefNode m8 = {
        sRel = {-0.08, 0, 0};
    };

```

```

    AnyRefNode m7 = {
        sRel = {-0.08, 0, 0};
    };

```

```

    AnyRefNode m3 = {
        sRel = {0, 0, 0};
    };

```

```

    AnyRefNode m1 = {
        sRel = {0.1, 0, 0};
    };

```

```

    AnyRefNode m6 = {
        sRel = {-0.1, 0, 0};
    };
}; // S1

```

```

AnySeg S2 = {
    r0 = {0.3, 0, 0};
    Mass = 5.2;
    Jii = {0.001, 0.022, 0.022};
    AnyRefNode E = {
        sRel = {-0.15, 0, 0};
    };
    AnyRefNode m3 = {
        sRel = {-0.1, 0, 0};
    };
    AnyRefNode m1 = {
        sRel = {0.1, 0, 0};
    };
};

```

```

    AnyRefNode m2 = {
        sRel = {0,0,0};
    };
    AnyRefNode m4 = {
        sRel = {-0.05,0,0};
    };
    AnyRefNode m5m6 = {
        sRel = {-0.20,0,0};
    };
    AnyDrawSeg drw = {
        Opacity = 0.4;
        InertiaScale = 0.5;
    };

}; // S2

}; // Segs folder

AnyFolder Jnts = {

    //-----
    AnyRevoluteJoint S = {
        Axis = z;
        AnyRefNode &GroundNode = ..GlobalRef.S;
        AnyRefNode &S1Node = ..Segs.S1.S;
    }; // Shoulder joint

    AnyRevoluteJoint E = {
        Axis = z;
        AnyRefNode &S1Node = Main.ArmModel.Segs.S1.E;
        AnyRefNode &S2Node = Main.ArmModel.Segs.S2.E;
    }; // Elbow joint

}; // Jnts folder

AnyFolder Drivers = {

    //-----
    AnyKinEqSimpleDriver ShoulderMotion = {
        AnyRevoluteJoint &Jnt = ..Jnts.S;
        DriverPos = {-110*pi/180};
        DriverVel = {30*pi/180};
        Reaction.Type = {Off};
//         DriverVel = {0}; // "rigifying" the shoulder
//         Reaction.Type = {On}; // "rigifying" the shoulder
    }; // Shoulder driver

    //-----
    AnyKinEqSimpleDriver ElbowMotion = {
        AnyRevoluteJoint &Jnt = ..Jnts.E;
        DriverPos = {70*pi/180};
        DriverVel = {45*pi/180};
        Reaction.Type = {Off};
    }; // Elbow driver
}; // Driver folder

AnyFolder Muscles = {

```

```

// Simple muscle model with constant strength
AnyMuscleModel MusMdl1 = {
    F0 = 0.75*150;
};
AnyMuscleModel MusMdl2 = {
    F0 = 0.75*300;
};
AnyMuscleModel MusMdl35 = {
    F0 = 0.75*500;
};
AnyMuscleModel MusMdl4 = {
    F0 = 0.75*750;
};
AnyVar p = 2; // used for poly, p=1-5
// AnyVar p = 1e6; // used to get F0=0.5*strength of m4
AnyMuscleModel MusMdl412 = {
    F0 = 0.75*750*2^((1-.p)/.p);
};
AnyMuscleModel MusMdl6 = {
    F0 = 0.75*900;
};
AnyMuscleModel MusMdl78 = {
    F0 = 0.75*1500;
};

//-----
AnyViaPointMuscle m1 = {
    AnyMuscleModel &MusMdl = ..Muscles.MusMdl1;
    AnyRefNode &Org = ..Segs.S1.m1;
    AnyRefNode &Ins = ..Segs.S2.m1;
    AnyDrawMuscle DrwMus = {MaxStress = 2500000};
};

//-----
AnyViaPointMuscle m2 = {
    AnyMuscleModel &MusMdl = ..Muscles.MusMdl2;
    AnyRefNode &Org = ..Segs.S1.m2;
    AnyRefNode &Ins = ..Segs.S2.m2;
    AnyDrawMuscle DrwMus = {MaxStress = 2500000};
};

//-----
AnyViaPointMuscle m3 = {
    AnyMuscleModel &MusMdl = ..Muscles.MusMdl35;
    AnyRefNode &Org = ..Segs.S1.m3;
    AnyRefNode &Ins = ..Segs.S2.m3;
    AnyDrawMuscle DrwMus = {MaxStress = 2500000};
};

//-----
// AnyViaPointMuscle m4 = {
//     AnyMuscleModel &MusMdl = ..Muscles.MusMdl4;
//     AnyRefNode &Org = ..GlobalRef.m4;
//     AnyRefNode &Ins = ..Segs.S2.m4;
//     AnyDrawMuscle DrwMus = {MaxStress = 2500000};
// };
//-----

```

```

AnyViaPointMuscle m41 = {
  AnyMuscleModel &MusMdl = ..Muscles.MusMdl412;
  AnyRefNode &Org = ..GlobalRef.m4;
  AnyRefNode &Ins = ..Segs.S2.m4;
  AnyDrawMuscle DrwMus = {MaxStress = 2500000;};
};
//-----
AnyViaPointMuscle m42 = {
  AnyMuscleModel &MusMdl = ..Muscles.MusMdl412;
  AnyRefNode &Org = ..GlobalRef.m4;
  AnyRefNode &Ins = ..Segs.S2.m4;
  AnyDrawMuscle DrwMus = {MaxStress = 2500000;};
};

//-----
AnyViaPointMuscle m5 = {
  AnyMuscleModel &MusMdl = ..Muscles.MusMdl35;
  AnyRefNode &Org = ..GlobalRef.m5;
  AnyRefNode &Ins = ..Segs.S2.m5m6;
  AnyDrawMuscle DrwMus = {MaxStress = 2500000;};
};

//-----
AnyViaPointMuscle m6 = {
  AnyMuscleModel &MusMdl = ..Muscles.MusMdl6;
  AnyRefNode &Org = ..Segs.S1.m6;
  AnyRefNode &Ins = ..Segs.S2.m5m6;
  AnyDrawMuscle DrwMus = {MaxStress = 2500000;};
};

//-----
AnyViaPointMuscle m7 = {
  AnyMuscleModel &MusMdl = ..Muscles.MusMdl78;
  AnyRefNode &Org = ..GlobalRef.m7;
  AnyRefNode &Ins = ..Segs.S1.m7;
  AnyDrawMuscle DrwMus = {MaxStress = 2500000;};
};

//-----
AnyViaPointMuscle m8 = {
  AnyMuscleModel &MusMdl = ..Muscles.MusMdl78;
  AnyRefNode &Org = ..GlobalRef.m8;
  AnyRefNode &Ins = ..Segs.S1.m8;
  AnyDrawMuscle DrwMus = {MaxStress = 2500000;};
};
}; // Muscles folder

AnyForceMomentMeasure ShoulderReaction = {
  AnyForceBase &vad = Main.ArmModel.Jnts.S.Constraints.Reaction;
  AnyRefFrame &var = Main.ArmModel.GlobalRef;
};
AnyForceMomentMeasure ElbowReaction = {
  AnyForceBase &vad = Main.ArmModel.Jnts.E.Constraints.Reaction;
  AnyRefFrame &var = Main.ArmModel.Segs.S1.E;
};
}; // MyModel

```

```
// The study: Operations to be performed on the model
AnyBodyStudy ArmStudy = {
  AnyFolder &Model = .ArmModel;
  Gravity = {0.0, -9.81, 0.0};
  InverseDynamics.Criterion = {
//    Type = MR_Linear; //poly with power 1
    Type = MR_Quadratic; //poly with power 2
//    Type = MR_Polynomial; Power = 3;
//    Type = MR_Polynomial; Power = 4;
//    Type = MR_Polynomial; Power = 5;
//    Type = MR_MinMaxStrict;
  };

};

}; // Main
```