Exploiting Dynamics Parameter Linearity for Design Optimization in Combined Structural and Dimensional Robot Synthesis

Moritz Schappler, Svenja Tappe, and Tobias Ortmaier

Institute for Mechatronic Systems, Leibniz University Hannover, Germany,
moritz.schappler@imes.uni-hannover.de

Abstract. In the design optimization of robot manipulators regarding drive train and link geometries the dynamics equations have to be evaluated repeatedly. The method proposed in this paper reduces the computational effort in the dynamics evaluations by using the property of parameter linearity of the dynamics equations. The combined structural and dimensional synthesis of robot manipulators is adapted in a set of hierarchical optimization loops to exploit this dynamics property. By this means a reduction of computation time for the inverse dynamics in the synthesis of up to factor three is possible.

Keywords: Robot Design Optimization, Dynamics Regressor Form, Combined Structural and Dimensional Synthesis

1 Introduction and State of the Art

Design optimization formalizes the process of designing technical systems done by engineers and allows to find values for the design parameters with optimization algorithms based on given objectives and constraints. In the case of robotic systems the design parameters can be the lengths of the robot links, the shape of the links or the selection of the drive train components motor and gearbox. The parameters can be deduced from the robot structure which consists of the type and principal alignment of the robots joints and is determined in the *structural synthesis* of the robot.

In this paper we distinguish between the optimization of parameters with influence on the kinematics and the optimization of parameters only influencing other characteristics of the robot such as the dynamics. The optimization of kinematic parameters is referred to as *dimensional synthesis* and can be implemented as multi-criterial optimization of kinematic criteria like workspace and robot size [1], static criteria like stiffness [2] or dynamics-based criteria like energy consumption [3]. A combined optimization of parameters influencing kinematics and dynamics can be performed by using parametric CAD models [4, 5] or by combined optimization of link and drive train properties [6].

For the *optimal selection of drive train components* separate from the kinematics parameters, these can be regarded as given. Common objective functions are the drive train mass [7–9], the trajectory cycle time [10], the predicted life time of the gears [11] or monetary costs [11]. Constraints contain at least the speed and torque limits of the components [7–9].

It is usually sufficient to regard the inverse rigid body dynamics without considering electric, thermal or control effects. The extension of the robot simulation within the optimization has been done for thermal dynamics in [7] and for electrical and controller dynamics in [12].

The *optimization of the parameters of the robot links* like the engineering design can be achieved with parametrized CAD-models prepared by a designer for the specific robot [10, 5] or by using geometric primitives [3] for approximations. The most straight-forward objective function for the link optimization consists of the masses of the links which also correlates with energy consumption and joint torques. Constraints can be implemented by performing stress analysis on the links CAD model using FEM programs like CATIA [10] or Ansys [5]. The load on the links and on the motors can be calculated using multibody programs like Modelica/Dymola [10], ADAMS [9] or by using optimized code from symbolic calculations [7] which reduces the computation [13].

The approaches presented until here can be separated into cases, where the trajectory does not change within the design optimization [10, 9, 12] and cases, where the trajectory is part of the optimization and only start and end pose of the end effector are given [11, 8, 10]. The trajectory changes within the optimization, if it is part of the objective function or if the kinematic parameters change in the optimization [4, 5], since the trajectory is usually given in task space and has to be transferred to joint space via the inverse kinematics.

For the discrete problem of selecting components of given gearbox and motor databases genetic algorithms [10, 12], particle swarm optimization [3] and the complex-algorithm with modifications regarding rounding and interpolation of the selection database [11, 9] can be used. Depending on the assumptions on the placement of actuators, a recursive joint-wise approach from distal to proximal joints is possible for serial link robots and simplifies the optimization [7].

The overview of the state of the art shows, that the field of design optimization for robot manipulators has made great progress in the last decade. One central element in the design optimization is the calculation of the robot dynamics, which should be implemented as efficient as possible. One method to improve the computation time spent on inverse dynamics calculations is to use the linearity of the dynamics equation with respect to the dynamics parameters. Regrouping the parameters to a minimal form allows to improve the performance even further [13], which is a key to the success of parameter identification experiments. To the best knowledge of the authors, the properties of the dynamics regressor form are not fully exploited by the design optimization community.

With the regressor form, it is possible to split the dynamics calculation into two steps: The calculation of the regressor and the multiplication of the regressor matrix with the dynamics parameter vector. Under certain assumptions the dynamics calculation in the design optimization process can then be performed with the second step. The first step can be done outside the optimization loop,

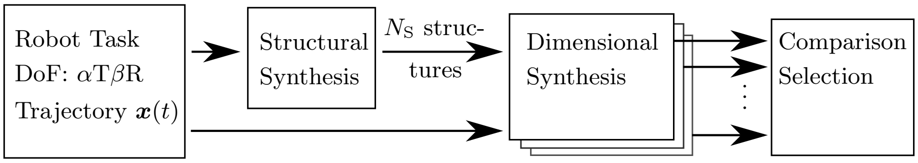

Fig. 1. Overview of the procedure for combined structural and dimensional synthesis. The degrees of freedom (DoF) of the task are given in the αTβR-notation for the numbers of translational DoF α and of rotational DoF β.

since the regressor matrix is dependent on the kinematics of the robot, which stays the same in the inner loop. The contributions of this paper are

- the presentation of an approach for a faster calculation of the inverse dynamics in the design optimization of robot manipulators,
- a concept for the combined structural and dimensional synthesis of serial and parallel robots exploiting this dynamics formulation,
- examples on the possible reduction of the computation time.

The remainder of the paper is structured as follows: The combined structural and dimensional synthesis is briefly presented as a motivation and field of application for the proposed method in Sec. 2. The properties of the dynamics model are summed up in Sec. 3. The possible improvement of the design optimization is explained in Sec. 4 with an estimate on the computational costs in Sec. 5.

2 Combined Structural and Dimensional Synthesis

The synthesis of robot manipulators can be divided in two phases. The first phase is finding the number, kind and alignment of joints in the *structural synthesis*. To obtain a result with quantifiable performance, the dimensioning of the parameters regarding actuation, gears, link lengths and link geometry have to be defined in the *dimensional synthesis* as a second phase. Usually the two phases are performed separately and the choice of a structure is done by the designer. Combining both phases means to perform the dimensional synthesis for all possible kinematic structures for a task with given degrees of freedom and trajectory as sketched in Fig. 1 for the whole process. This allows a comparison of all N_S different structures with optimized dimensions for the given task and to select the robot structure that is suited best [14, 3].

Despite the existence of methods to reduce the number of kinematic structures from the structural synthesis to a minimum [14], the computational effort in the dimensional synthesis is still high. For serial chain robots e. g. with the assumption of non-redundancy and only parallel and orthogonal joints there exist N_S=10 possible configurations for planar motions (2T1R), N_S=35 for Schönflies motions (3T1R, "SCARA") and N_S=326 for general motions (3T3R) [14, 3]. An even higher number of parallel robot structures exists due to the possibility of overconstraint, non-symmetry and selection of joint actuation, which are listed in [15] and its following parts.

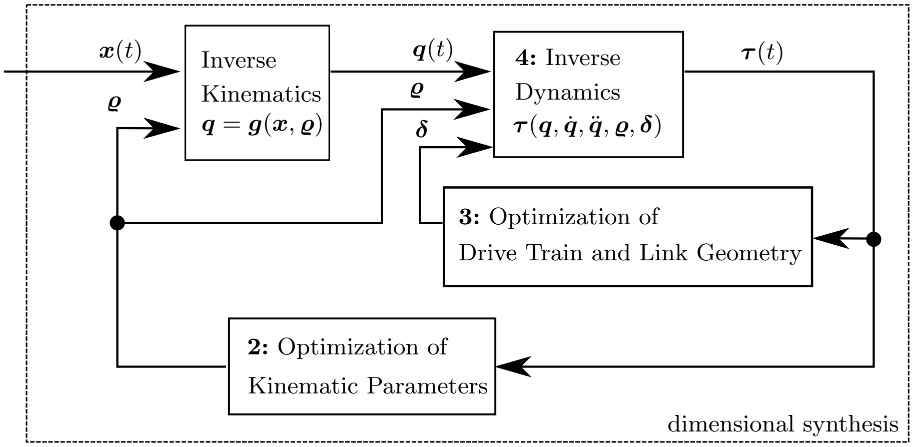
$\boldsymbol{x}(t)$
$\boldsymbol{\varrho}$
Inverse Kinematics $\boldsymbol{q} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\varrho})$
$\boldsymbol{q}(t)$
$\boldsymbol{\varrho}$
$\boldsymbol{\delta}$
4: Inverse Dynamics $\boldsymbol{\tau}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\varrho}, \boldsymbol{\delta})$
$\boldsymbol{\tau}(t)$
3: Optimization of Drive Train and Link Geometry
2: Optimization of Kinematic Parameters
dimensional synthesis

Fig. 2. Overview of the placement of inverse dynamics calculation in the nested optimization loops of dimensional synthesis.

The high computational effort required for the combined synthesis is caused by the multiplicity of optimization loops, namely from outside to inside

1. a loop over all N_S possible robot structures,
2. the optimization of the kinematic parameters $\boldsymbol{\varrho}$ (length of links),
3. the design optimization of the drive train and the link geometry affecting the dynamics parameters $\boldsymbol{\delta}$ and requiring the dynamics calculation in
4. a loop over all N_T sample points of the given trajectory $\boldsymbol{x}(t)$.

The loops 2 to 4 are summarized in Fig. 2, which depicts the nested optimization loops in a block diagram that corresponds to the dimensional synthesis block of Fig. 1. There, the first loop is highlighted by the depicted stacked blocks. The time dependent quantities end effector pose $\boldsymbol{x}(t)$, joint coordinates $\boldsymbol{q}(t)$ and joint torque $\boldsymbol{\tau}(t)$ are assumed to contain the full time series and the needed time derivatives over all N_T samples of the trajectory.

When using only kinematic performance measures like workspace size or dexterity for the dimensional synthesis, the dynamics do not have to be evaluated and the third loop is not necessary, which simplifies the optimization. On the other hand, when using dynamics performance measures or when the decision for a robot structure also incorporates the possibilities of the actuation, all loops have to be taken into account.

To reduce the computational effort, the trajectory $\boldsymbol{x}(t)$ is regarded as given and not as part of the optimization. This does not allow to use the point-to-point cycle time as a performance measure. On the upside, the comparison of different structures like serial and parallel robots is reduced in complexity and it allows to use the method proposed in the latter part of the paper. The demand for the reduction of complexity can be justified by the fact, that serial and parallel robots have completely different characteristics regarding their performance criteria like workspace, footprint, accuracy, stiffness, energy or dynamics. These criteria have to be traded off against each other in a single-objective or multi-objective optimization. Comparing the dynamics or energy for different structures each with different time-series of the trajectory would again raise the question of how to weight these performance measures against each other.

3 Properties of the Dynamics Model

The inverse dynamics equations are used in the design optimization of robot links and drive trains as can be seen in Fig. 2. Its purpose is to calculate the torque on the motor and the forces and moments on the links to evaluate if the considered configuration is feasible. Additionally, the properties of motor, gearbox and link affect the dynamics parameters and therefore the result of the equations. Since this paper focuses on the rigid body dynamics, other effects such as friction are neglected and the quantities of the drive train like motor torque, motor inertia and gear ratio are assumed to be included in the joint space equations and are omitted in the following. This section defines the dynamics model in a nomenclature suitable to show the improvement of the design optimization in the next section and corresponds to textbook-knowledge [13].

To investigate the influence of the kinematic and dynamics parameters, the parameters are written explicitly in the equations. The kinematics parameters $\boldsymbol{\varrho}$ contain the parameters from the joint transformations according to the modified Denavit-Hartenberg notation and are e. g. the lengths of the links. The dynamics parameters are defined in three representations:

- The barycentric dynamics parameters of each link i consist of the link mass m_i, the three coordinates ${}_{(i)}\boldsymbol{r}_{C_i}$ of the links center of mass C_i (CoM) expressed in the respective link frame and six entries of the symmetric inertia tensor ${}_{(i)}\boldsymbol{I}^{(C_i)}$ w. r. t the links CoM.
- The inertial parameters are related to the origin O_i of the link frame and contain the links mass m_i, the first moment $m_{i(i)}\boldsymbol{r}_{C_i}$ and the second moment ${}_{(i)}\boldsymbol{I}^{(O_i)} = {}_{(i)}\boldsymbol{I}^{(C_i)} + m_i \boldsymbol{S}^{\mathrm{T}}({}_{(i)}\boldsymbol{r}_{C_i})\boldsymbol{S}({}_{(i)}\boldsymbol{r}_{C_i})$ which can be calculated with the Huygens-Steiner theorem using the skew-operator $\boldsymbol{S}$. The 10 inertial parameters for link i are stacked in $\boldsymbol{\delta}_i$ and the $10N$ parameters for all N links are stacked in the vector $\boldsymbol{\delta}$.
- The base inertial parameters can be obtained by combining the inertial parameters with a set of rules to the linear relation $\boldsymbol{\beta} = \boldsymbol{K}(\boldsymbol{\varrho})\boldsymbol{\delta}$.

The joint torques $\boldsymbol{\tau}$ of a serial-chain robot manipulator are calculated with the inverse dynamics model

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{M}(\boldsymbol{q},\boldsymbol{\varrho},\boldsymbol{\delta})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\varrho},\boldsymbol{\delta}) + \boldsymbol{g}(\boldsymbol{q},\boldsymbol{\varrho},\boldsymbol{\delta}) = \boldsymbol{\tau}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},\boldsymbol{\varrho},\boldsymbol{\delta}) \qquad (1)\\ &= \boldsymbol{\Phi}_{\boldsymbol{\tau}}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},\boldsymbol{\varrho})\boldsymbol{\delta}, \qquad (2)\\ \tau_i &= \boldsymbol{\Phi}_{\tau_i}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},\boldsymbol{\varrho})\boldsymbol{\delta} \qquad (3)\end{aligned}$$

in the joint coordinates $\boldsymbol{q}$. It contains the effects of inertia $\boldsymbol{M}$, centrifugal/Coriolis forces $\boldsymbol{c}$ and gravity $\boldsymbol{g}$ and is linear in the inertial parameters $\boldsymbol{\delta}$ with the regressor matrix $\boldsymbol{\Phi}_{\boldsymbol{\tau}}$.

By deriving the dynamics with the Newton-Euler-algorithm, it is also possible to express the internal forces $\boldsymbol{f}_i$ and moments $\boldsymbol{m}_i$ in the joint i in the parameter linear form

$$\boldsymbol{w}_i = \left(\boldsymbol{f}_i^{\mathrm{T}}\ \boldsymbol{m}_i^{\mathrm{T}}\right)^{\mathrm{T}} = \boldsymbol{\Phi}_{\boldsymbol{w}_i}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\varrho})\boldsymbol{\delta}. \tag{4}$$

The regressor matrices $\boldsymbol{\Phi}_{\boldsymbol{\tau}}$ and $\boldsymbol{\Phi}_{\boldsymbol{w}}$ have the upper triangular form

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\Phi}_1 \\ \boldsymbol{\Phi}_2 \\ \vdots \\ \boldsymbol{\Phi}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Phi}_{1,\boldsymbol{\delta}_1} & \boldsymbol{\Phi}_{1,\boldsymbol{\delta}_2} & \dots & \boldsymbol{\Phi}_{1,\boldsymbol{\delta}_N} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{2,\boldsymbol{\delta}_2} & \dots & \boldsymbol{\Phi}_{2,\boldsymbol{\delta}_N} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_{N,\boldsymbol{\delta}_N} \end{pmatrix}, \tag{5}$$

where the influence of the inertial parameters of link j on joint i is marked by $\boldsymbol{\Phi}_{i,\boldsymbol{\delta}_j}$ and the row $\boldsymbol{\Phi}_i$ associated to one joint corresponds to $\boldsymbol{\Phi}_{\tau_i}$ from (3) or $\boldsymbol{\Phi}_{\boldsymbol{w}_i}$ from (4). This shows the property of serial kinematic chains, that the inertial parameters of links proximal to the robot base have no influence on distal joints. For example the last wrist joint of an N-DoF industrial robot only articulates the mass and inertia of the flange and the end-effector, which are summed up in $\boldsymbol{\delta}_N$ and correspond to the last line of (5). Other masses do not affect τ_N or $\boldsymbol{w}_N$

The dynamics equation (1-2) can also be expressed with the base[1] inertial parameters $\boldsymbol{\beta}$. These parameters are obtained with a recursive combination of inertial parameters with the same effect on the joint torque. Parameters without effect are removed from the parameter vector. Due to the recursive scheme of grouping parameters from the end-effector link to the base link, the regressor matrix with respect to the base inertial parameters has a similar upper triangular form as in (5). Exploiting this known structure allows further reduction of the computational load.

The joint torque equation (1) can be used for design optimization of the drive train. For the optimization of link parameters, the full internal forces and moments from (4) should be used. These forces have to be expressed with the complete inertial parameters $\boldsymbol{\delta}$ and not with the base parameters $\boldsymbol{\beta}$, since all inertial parameters affect the internal forces and no parameter reduction is possible. In the following, the regressor with respect to the inertial parameters $\boldsymbol{\delta}$ will be used in this paper for the sake of simplicity. The base parameter regressor should however be used in the implementation wherever possible, since it is more efficient.

The equations and remarks above regarding the sparsity of the triangular form of the regressor matrix refer to serial robots and are not necessary for the proposed method. The dynamics of parallel robots can also be expressed in a regressor form, since it can be derived by projection of the legs and platform dynamics into the task space. Grouping the parameters to a set of base inertial parameters again allows the reduction of the complexity. The scheme described in the following section can therefore also be applied to parallel robots.

[1] "Base" refers to the minimal set of parameters in the mathematical sense.

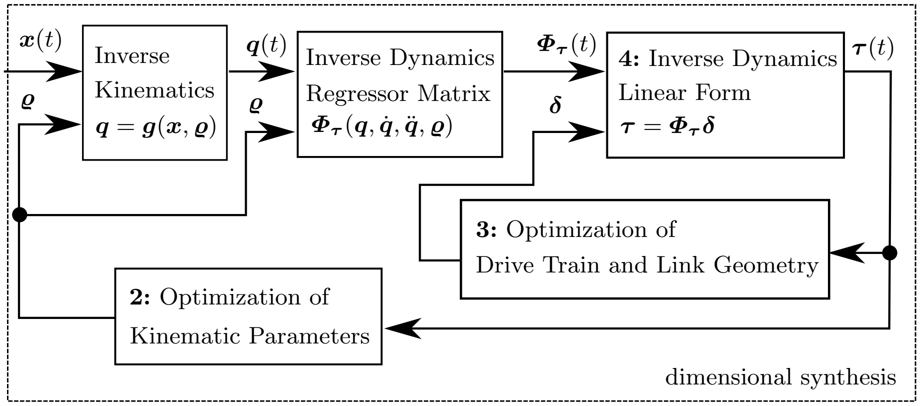
$\boldsymbol{x}(t)$
$\boldsymbol{\varrho}$
Inverse Kinematics $\boldsymbol{q} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\varrho})$
$\boldsymbol{q}(t)$
$\boldsymbol{\varrho}$
Inverse Dynamics Regressor Matrix $\boldsymbol{\Phi}_{\boldsymbol{\tau}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\varrho})$
$\boldsymbol{\Phi}_{\boldsymbol{\tau}}(t)$
$\boldsymbol{\delta}$
4: Inverse Dynamics Linear Form $\boldsymbol{\tau} = \boldsymbol{\Phi}_{\boldsymbol{\tau}}\boldsymbol{\delta}$
$\boldsymbol{\tau}(t)$
3: Optimization of Drive Train and Link Geometry
2: Optimization of Kinematic Parameters
dimensional synthesis

Fig. 3. Sketch of the improved calculation of dynamics in the nestested optimization loops of dimensional synthesis. The numbers of the loops are referenced in the text.

4 Improve the Efficiency of Design Optimizing

With the assumptions from Sec. 2 of an unchangeable task space trajectory and a separate optimization of kinematic and dynamics parameters, the presented properties of the dynamics equations from Sec. 3 can be used to accelerate the evaluation of the dynamics in the dimensional synthesis.

Since the inverse kinematics can be calculated in the second loop (over the kinematics parameters $\boldsymbol{\varrho}$) and the dynamics equations are evaluated in a third loop (of the drive train optimization updating $\boldsymbol{\delta}$), the joint space trajectory $\boldsymbol{q}(t)$ can be regarded as given in the third loop. Then, as depicted in Fig. 3, the property (2) of linearity of the dynamics equation can be used to divide it into the calculation of the regressor matrices $\boldsymbol{\Phi}(t)$ in the second loop and the multiplication of the regressor with the dynamics parameter vector $\boldsymbol{\delta}$ in the third loop. The regressor $\boldsymbol{\Phi}_{\boldsymbol{w}}$ for the full internal forces $\boldsymbol{w}$ from (4) can be used in the same manner and is omitted in Fig. 3 for clarity.

5 Exemplary Calculations

Using the linearity of dynamics parameters as described above reduces the computational cost of evaluating the dynamics equations down to the multiplication of the regressor matrix with the parameter vector. Table 1 gives an estimate on the computational cost for the dynamics of three different serial chain robots: A 4-DoF SCARA robot, a 6-DoF industrial robot arm manipulator and the 7-DoF DLR/Kuka Light Weight Robot (LWR). The upper part of Table 1 shows the properties of the robots kinematic and dynamics parameters (rows “A”-“D”). The lower part summarizes the computational cost of the dynamics using direct

Table 1. Overview of the parameter properties and the computational effort for inverse dynamics calculation for three serial robot examples.

	Robot	SCARA	Ind. Robot	LWR
A	dim($\boldsymbol{q}$): number (#) of joint DoF	4	6	7
B	dim($\boldsymbol{\varrho}$): # of kinematic parameters	2	6	3
C	dim($\boldsymbol{\delta}$): # of inertial parameters	40	60	70
D	dim($\boldsymbol{\beta}$): # of base inertial parameters	10	38	45
E	ratio of zeros in the $\boldsymbol{\beta}$-regressor $\boldsymbol{\Phi_\tau}$	19/40 = 48 %	87/228 = 38 %	124/315 = 39 %
F	# of operations for one regressor evaluation	327	1796	3003
G	# of operations for one dynamics evaluation	328	2078	3385
H	# of operations for regressor multiplication	42	282	382
I	decrease using regressor multiplication	87 %	86 %	89 %

calculation from the right-hand side of (1) in one case (row "G") and solely the regressor multiplication of (2) in the second case (row "H"). Both calculations are implemented using the base parameter regressor form and all possible optimizations to ensure a fair comparison.

The dynamics equations are implemented as Matlab functions, which are compiled for faster execution with complete trajectories as input to reduce overhead. The code is generated by the computer algebra system Maple using generalized joint coordinates, kinematics based on homogenous transformation matrices, a formulation of the energy of the multy body system w.r.t. the inertial dynamics parameters, the Lagrangian equations of the second kind and a base parameter regrouping algorithm, see [13]. Code optimization is done by Maple, also exploiting the sparsity of the triangular form (5) of the regressor, which can be recognized from the ratios of zeros given in table row "E". The number of operations in the table contains the unweighted sum of the number of additions, multiplications and assignments of temporary variables.

Since the calculation using a given regressor matrix is much simpler, the computational effort can be decreased in theory over 80 %, as highlighted in table row "I". However, some aspects reduce the savings in the practical implementation:

- Saving and accessing the dim($\boldsymbol{q}$) × dim($\boldsymbol{\beta}$) regressor matrices for all N_T trajectory samples leads to additional overhead in the memory access,
- the regressor matrix has to be computed beforehand, which will pay off more, if the dynamics are evaluated very often in the inner optimization loop.

The reduction of computation time is evaluated in a Matlab simulation at the example of one loop of the dimensional synthesis (block "2" in Fig. 2 and Fig. 3) for the three serial chain robots of Table 1. The computation time is compared for different numbers of dynamics parameter iterations in the design optimization of the actuators (block "3" in Fig. 3). The summed time for the calculation of the joint torque regressor matrix (5) and the time needed for the multiplication in (2) for all dynamics parameter iterations is referred to as "RegMult". The direct calculation of the inverse dynamics of (1) or of block "3"/"4" of Fig. 2 is marked as "InvDyn".

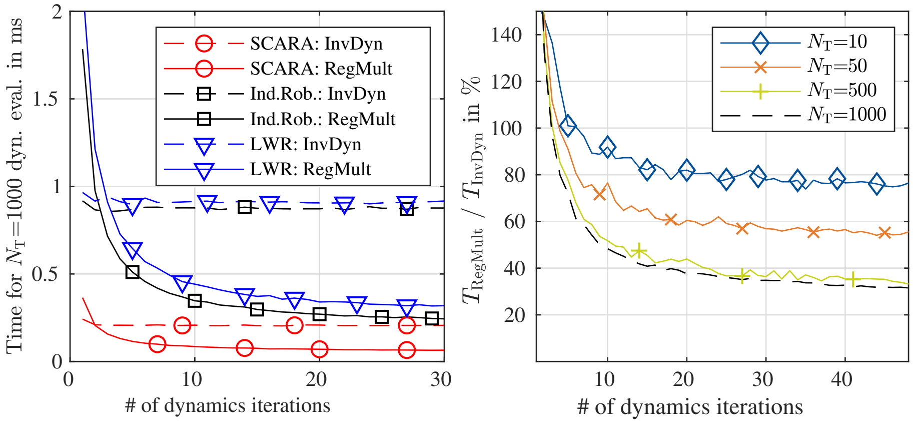

Fig. 4. Left: Comparison of the computation time for inverse dynamics for three robots and two methods. Right: Ratio of the computation times for different trajectory lengths.

The left part of Fig. 4 shows the computation time for the inverse dynamics of N_{T}=1000 samples of a joint trajectory $\boldsymbol{q}(t)$. The "RegMult"-method (dashed lines) already outperforms the "InvDyn"-method at five or more iterations of the dynamics parameters. This can be explained by the low computational cost of the regressor matrix, listed in row "F" of Table 1, which is approximately as high as one dynamics evaluation.

The influence of the length of the trajectory is investigated with the right part of Fig. 4 at the example of the 7-DoF LWR. The results for the other robots are similar. The ratios of the computation times for "RegMult" and "InvDyn" are compared for different lengths N_{T} of the trajectory. The "RegMult"-method is faster, if the line is below 100 %. The method benefits from a higher number of trajectory samples, which converges to a time reduction of 70 % for $N_{\mathrm{T}} > 100$ and more than 50 dynamics iterations.

The good performance of the proposed method "RegMult" can be explained by the low influence of the increased memory load of accessing the stacked regressor matrices instead of the joint trajectory. The method loses its advantage if only very few samples N_{T} of a trajectory, e. g. few static poses, are evaluated.

6 Conclusions

This paper presents an optimization scheme to select robot manipulators for a given task in a combined structural and dimensional synthesis. The optimization required to find the best suitable manipulator consists of nested loops of selecting different structures, optimizing kinematic parameters and optimizing dynamics parameters via design optimization of drive train and link properties. The loops are arranged in a way to improve the efficiency of the whole process by exploiting the parameter linearity of the dynamics equations.

Acknowledgements

The financial support from the Deutsche Forschungsgemeinschaft (DFG) under grant number OR 196/33-1 is gracefully acknowledged.

References

1. Ceccarelli, M., Lanni, C.: A multi-objective optimum design of general 3R manipulators for prescribed workspace limits. Mechanism and Machine Theory **39**(2), 119–132 (2004)
2. Carbone, G., Ottaviano, E., Ceccarelli, M.: An optimum design procedure for both serial and parallel manipulators. Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science **221**(7), 529–843 (2007)
3. Ramirez, D., Kotlarski, J., Ortmaier, T.: Combined structural-dimensional synthesis of robot manipulators for minimal energy consumption. In: Tagungsband des 2. Kongresses Montage Handhabung Industrieroboter, pp. 63–71. Springer (2017)
4. Tarkian, M., Lundén, B., Ölvander, J.: Integration of parametric cad and dynamic models for industrial robot design and optimization. In: ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 761–769
5. Zhou, L., Bai, S.: A new approach to design of a lightweight anthropomorphic arm for service applications. Journal of Mechanisms and Robotics **7**(3) (2015)
6. Shiller, Z., Sundar, S.: Design of robotic manipulators for optimal dynamic performance. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, pp. 334–339
7. Chedmail, P., Gautier, M.: Optimum choice of robot actuators. Journal of engineering for industry **112**(4), 361–367 (1990)
8. Pettersson, M., Ölvander, J.: Drive train optimization for industrial robots. IEEE Transactions on Robotics **25**(6), 1419–1424 (2009)
9. Zhou, L., Bai, S., Hansen, M.R.: Design optimization on the drive train of a lightweight robotic arm. Mechatronics **21**(3), 560–569 (2011)
10. Tarkian, M., Persson, J., Ölvander, J., Feng, X.: Multidisciplinary design optimization of modular industrial robots. In: Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE. Washington, DC (2011)
11. Pettersson, M., Andersson, J., Krus, P.: Methods for discrete design optimization. In: ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 295–303
12. Padilla-García, E.A., Cruz-Villar, C.A., Rodriguez-Angeles, A.: Multi-objective design/control optimization on the power train of robot manipulators using a genetic algorithm. In: Proc. of the 14th IFToMM World Congress (2015)
13. Khalil, W., Dombre, E.: Modeling, Identification and Control of Robots. Hermes Penton Science (2002)
14. Ramirez, D., Kotlarski, J., Ortmaier, T.: Automatic generation of a minimal set of serial mechanisms for a combined structural - geometrical synthesis. In: Proc. of the 14th IFToMM World Congress. Taipei (2015)
15. Gogu, G.: Structural Synthesis of Parallel Robots, Part 1: Methodology, *Solid Mechanics and Its Applications*, vol. 866. Springer (2008)